CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76001 CARLOS EDUARDO ANTAO
24FRO508 - 1 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

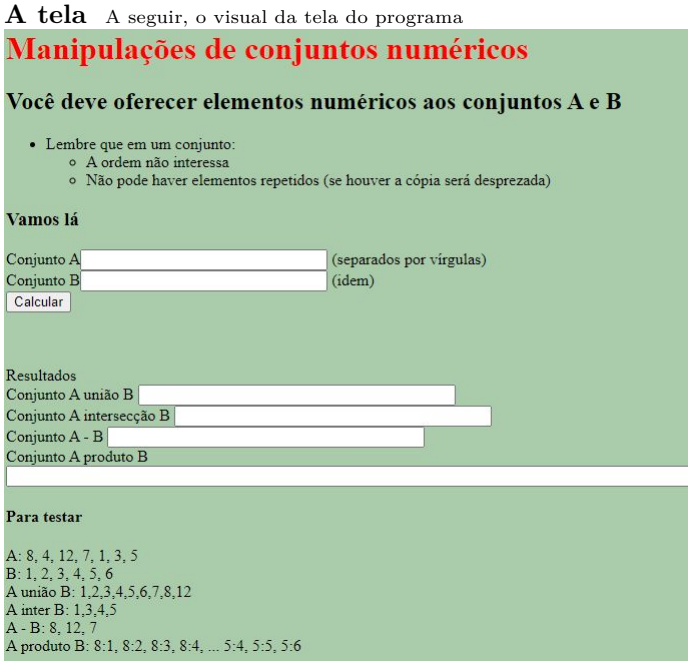
Agora execute o seu programa com os seguintes conjuntos

$A = \{9, 7, 5, 2, 8\}$
$B = \{5, 2, 8, 6, 7\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76001 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76199 DANILO DEPETRIS SOARES
24FRO508 - 2 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

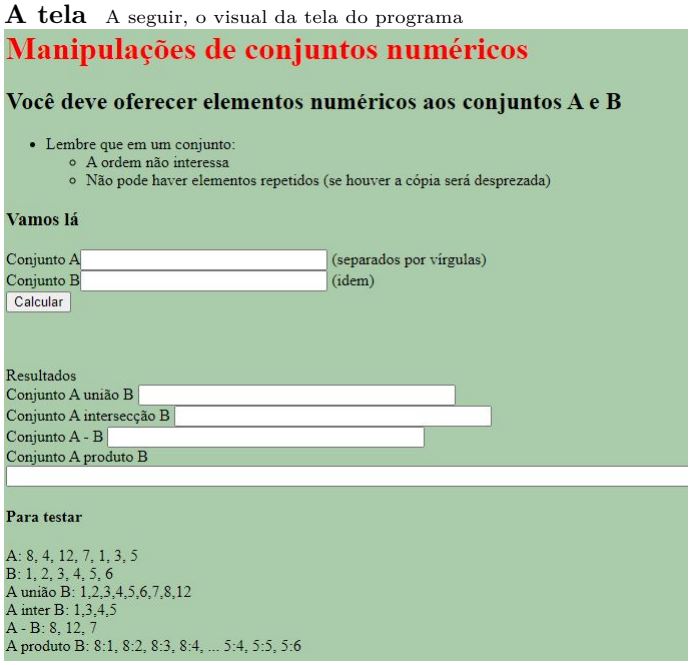
Agora execute o seu programa com os seguintes conjuntos

$A = \{2, 7, 5, 9, 8\}$
$B = \{8, 1, 7, 6, 4\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76199 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76018 EDUARDO CAVALHEIRO DIAS
24FRO508 - 3 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

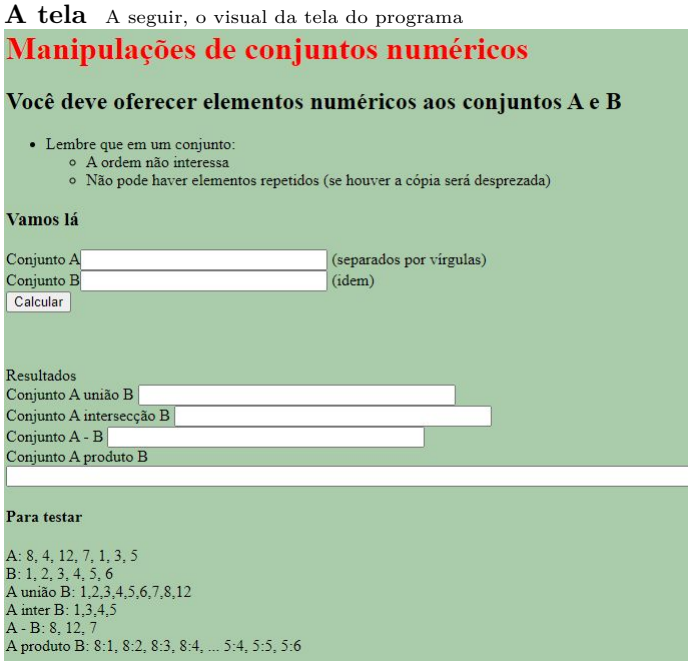
Agora execute o seu programa com os seguintes conjuntos

$A = \{9, 7, 6, 1, 3\}$
$B = \{2, 8, 9, 3, 4\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76018 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76025 ERIC CORDEIRO
24FRO508 - 4 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

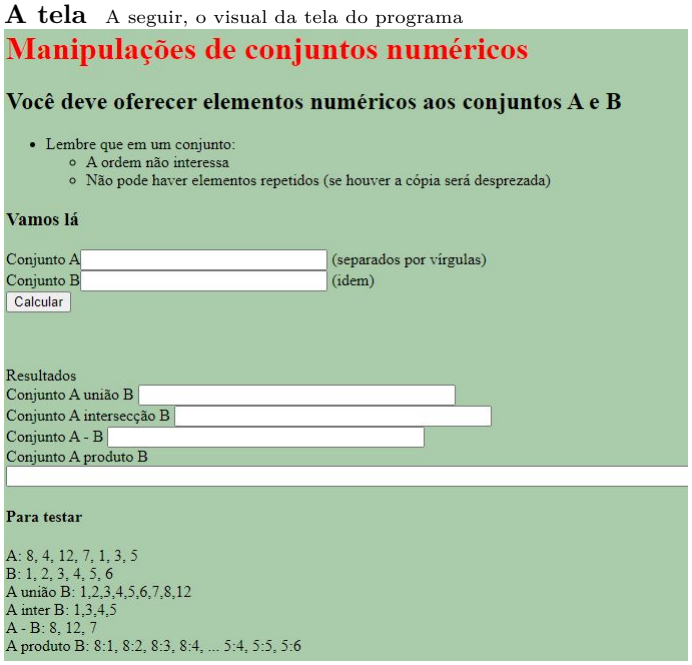
Agora execute o seu programa com os seguintes conjuntos

$A = \{4, 3, 5, 9, 7\}$
$B = \{7, 3, 6, 2, 1\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76025 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76032 FELIPE EIJI KANO
24FRO508 - 5 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

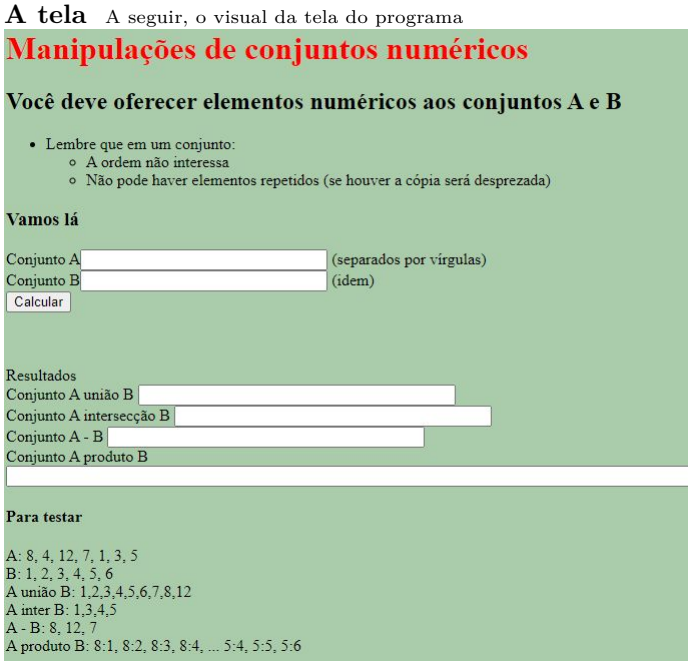
Agora execute o seu programa com os seguintes conjuntos

$A = \{7, 3, 4, 5, 1\}$
$B = \{3, 8, 9, 6, 7\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76032 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76049 FELIPE JESMIEL LEITE
24FRO508 - 6 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

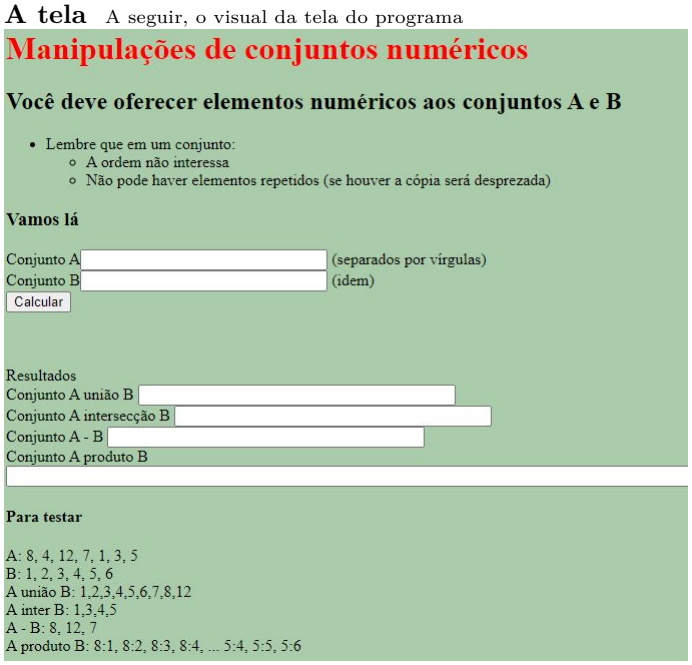
Agora execute o seu programa com os seguintes conjuntos

$A = \{5, 7, 8, 3, 1\}$
$B = \{3, 4, 5, 1, 9\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76049 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76056 GABRIELA BERNARDO PASQUALOTT
24FRO508 - 7 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

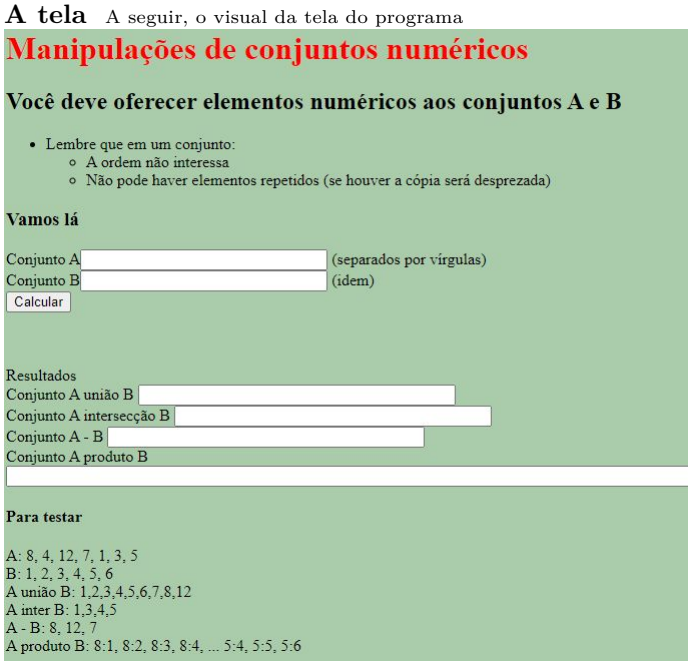
Agora execute o seu programa com os seguintes conjuntos

$A = \{2, 3, 1, 5, 8\}$
$B = \{5, 7, 6, 1, 4\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76056 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76720 GRAZIELLY A. L. VALICOSKI
24FRO508 - 8 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

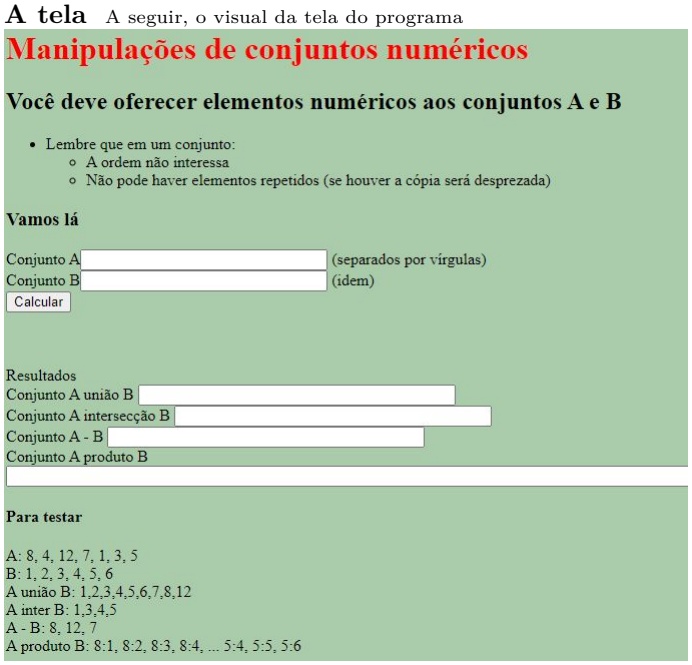
Agora execute o seu programa com os seguintes conjuntos

$A = \{5, 1, 4, 2, 7\}$
$B = \{5, 4, 3, 9, 1\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76720 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76063 GUILHERME VITOR DA SILVA
24FRO508 - 9 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

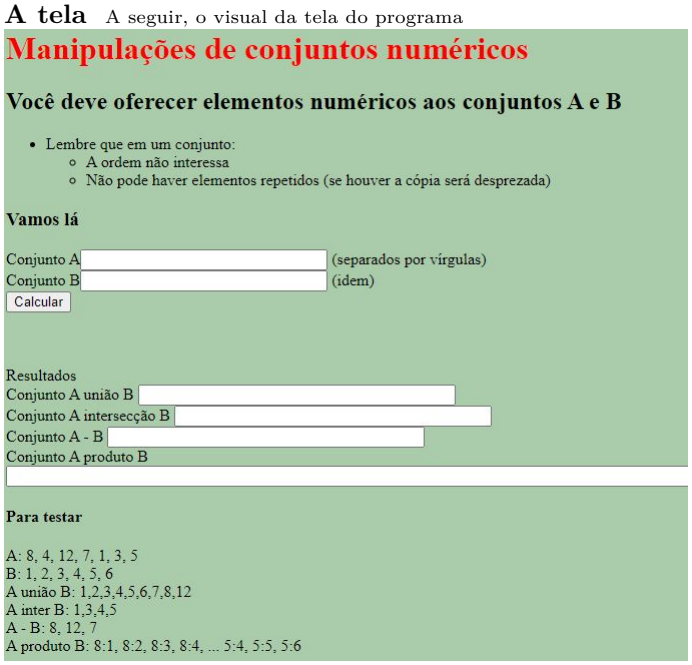
Agora execute o seu programa com os seguintes conjuntos

$A = \{3, 4, 9, 8, 1\}$
$B = \{1, 8, 3, 5, 9\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76063 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76713 KAWAN DAIKI HASEGAWA
24FRO508 - 10 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

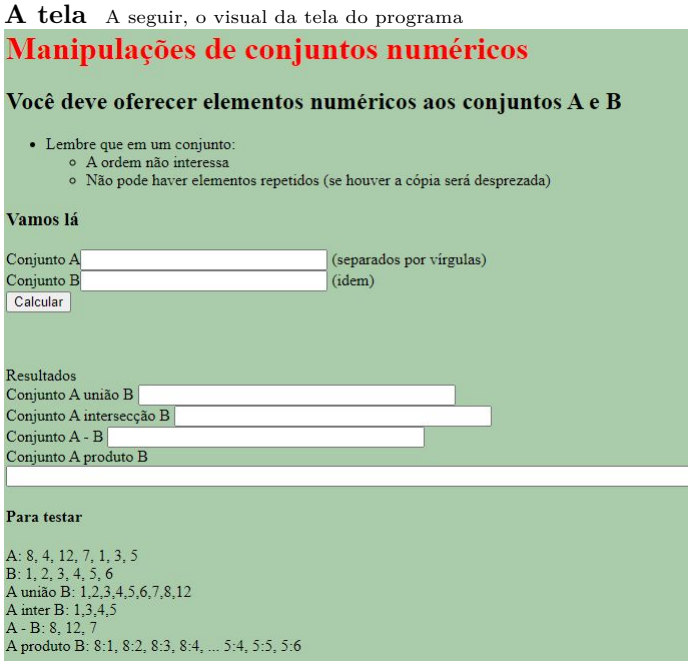
Agora execute o seu programa com os seguintes conjuntos

$A = \{7, 2, 5, 3, 8\}$
$B = \{3, 8, 6, 7, 5\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76713 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76087 LAURA PACHECO DELATTRE
24FRO508 - 11 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

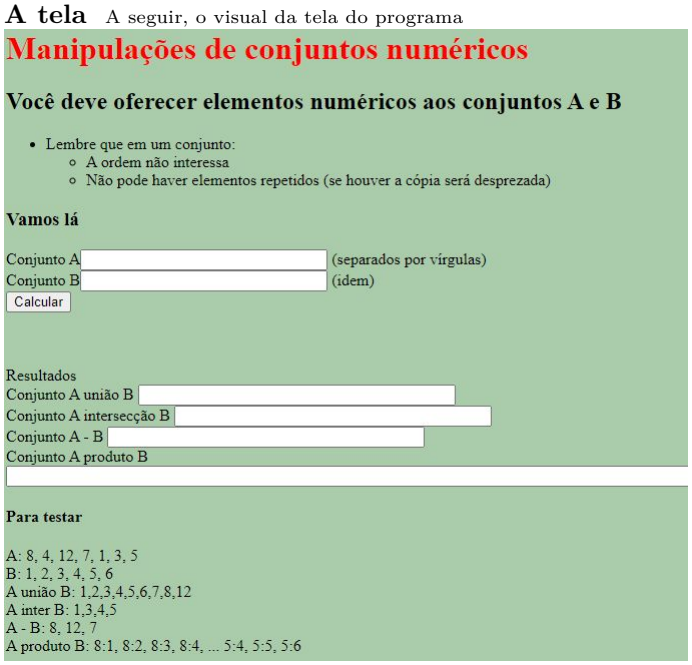
Agora execute o seu programa com os seguintes conjuntos

$A = \{8, 4, 7, 5, 1\}$
$B = \{9, 2, 1, 3, 5\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76087 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76687 LAURO HENRIK RUTZ
24FRO508 - 12 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

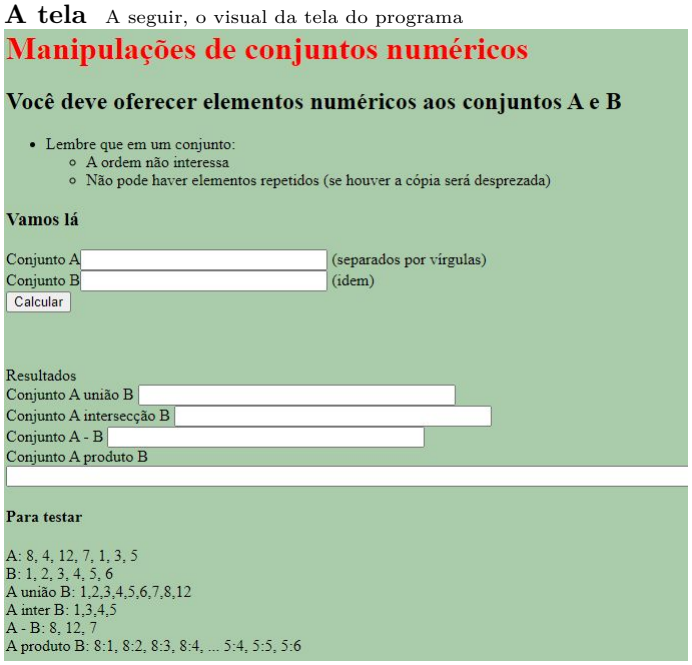
Agora execute o seu programa com os seguintes conjuntos

$A = \{9, 2, 8, 5, 4\}$
$B = \{1, 3, 5, 8, 4\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76687 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76106 LUIZ MIGUEL OLINEK
24FRO508 - 13 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

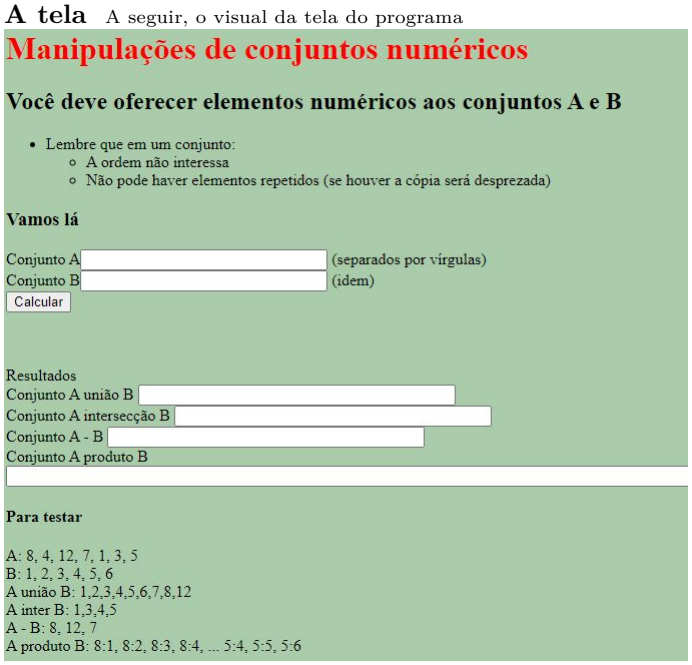
Agora execute o seu programa com os seguintes conjuntos

$A = \{3, 4, 8, 6, 7\}$
$B = \{7, 4, 2, 1, 6\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76106 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76113 NICOLE DA SILVA DE AZEVEDO
24FRO508 - 14 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

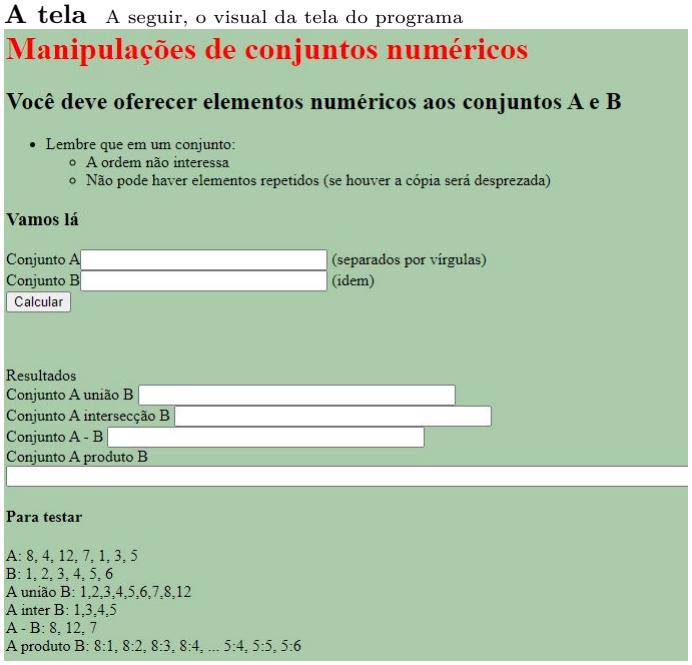
Agora execute o seu programa com os seguintes conjuntos

$A = \{9, 2, 7, 4, 8\}$
$B = \{9, 5, 1, 7, 6\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76113 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76120 PAULO JOSE FREIRE CORREA
24FRO508 - 15 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

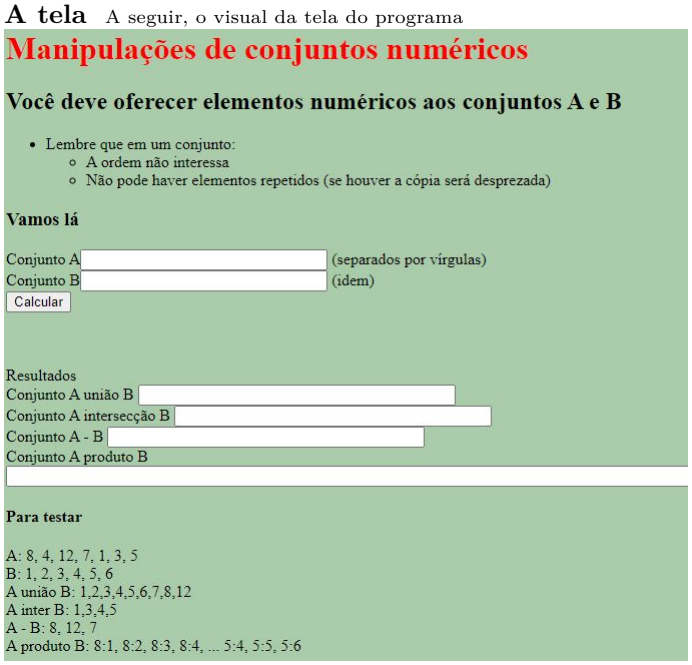
Agora execute o seu programa com os seguintes conjuntos

$A = \{1, 4, 8, 5, 3\}$
$B = \{6, 9, 5, 8, 4\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76120 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76137 PEDRO HENRIQUE PIEKARSKI DE 24FRO508 - 16 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

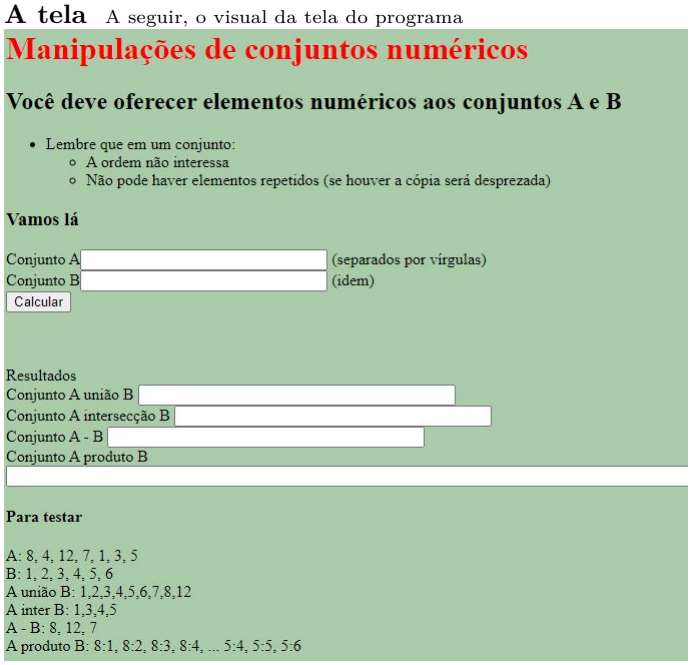
Agora execute o seu programa com os seguintes conjuntos

$A = \{4, 1, 2, 5, 3\}$
$B = \{6, 3, 7, 4, 2\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76137 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76144 RAUL KNAPKI CUNHA
24FRO508 - 17 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

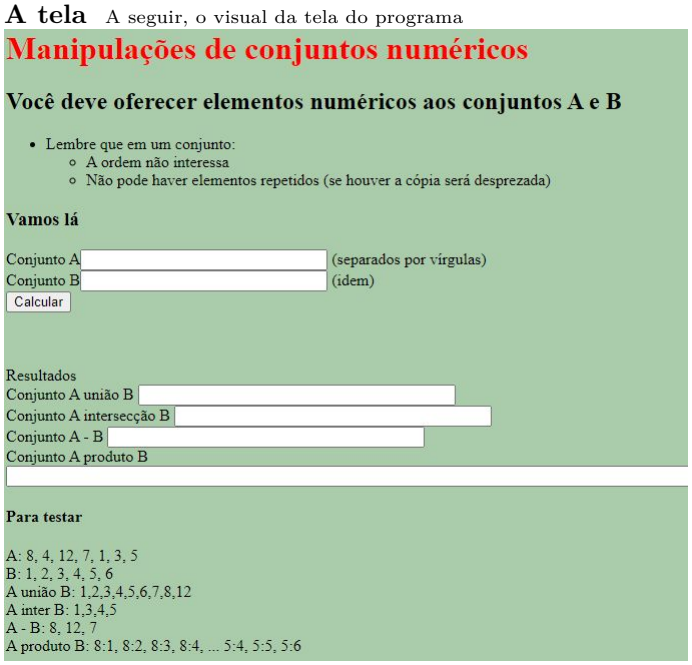
Agora execute o seu programa com os seguintes conjuntos

$A = \{4, 8, 1, 2, 3\}$
$B = \{9, 1, 4, 6, 8\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76144 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76168 VICTOR BRONDANI DOS SANTOS
24FRO508 - 18 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

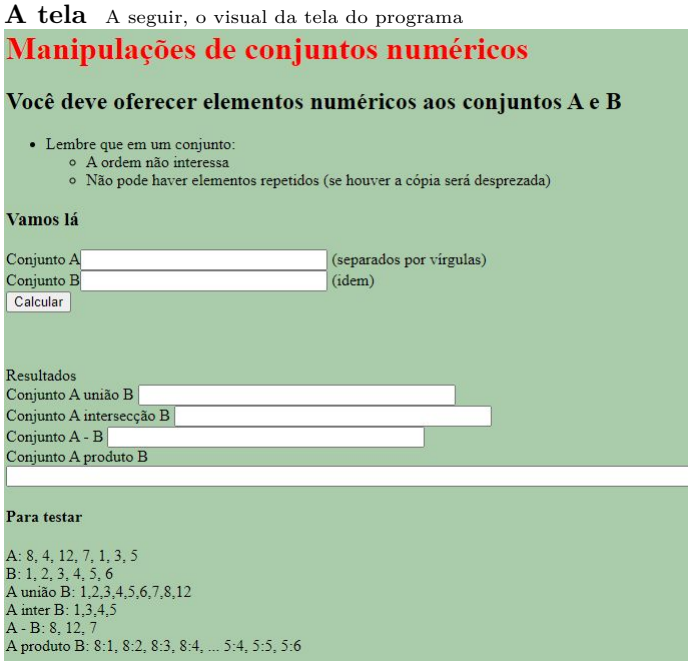
Agora execute o seu programa com os seguintes conjuntos

$A = \{1, 2, 4, 7, 3\}$
$B = \{3, 5, 2, 9, 1\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76168 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76175 VICTOR HUGO DOS SANTOS DE CA
24FRO508 - 19 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

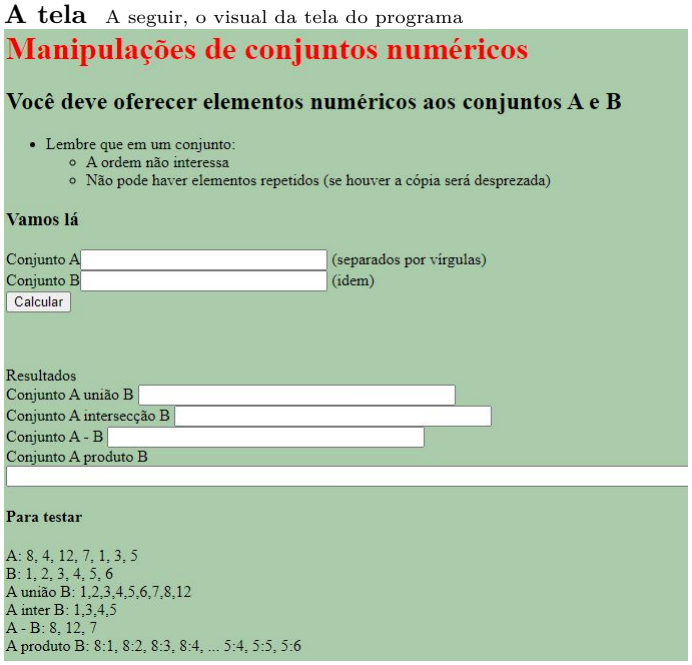
Agora execute o seu programa com os seguintes conjuntos

$A = \{4, 3, 8, 1, 2\}$
$B = \{5, 3, 6, 8, 7\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76175 - ga/ a

A tela A seguir, o visual da tela do programa



CEP-UFPR-UTFPR-PUC/Pr-UP Sistemas de Informação 27/10/2024 - 09:48:02.8
Matemática aplicada Prof Dr P Kantek (pkantek@gmail.com)
Conjuntos em JS VIVXh97j V: 1.02
76182 VITOR EDUARDO DE GOES FONTES
24FRO508 - 20 entregar ate 14/nov/24 _____ / / _____

Conjuntos em JS

Ao lidar com conjuntos em JS, pode-se lançar mão do objeto `Set` presente na linguagem ou, usando listas, implementar as funcionalidades associadas a conjuntos *na mão*. No material que segue existem as 2 coleções de funcão. As do primeiro tipo são `criarConjunto1`, `uniao1`, `intersecao1` e `diferenca1`. Já as do segundo tipo são `criarConjunto2`, `uniao2`, `intersecao2` e `diferenca2`. Ao programar esta folha, escolha qual quer implementar. Por óbvio, ambas implementações precisam dar o mesmo resultado.

A encomenda Escreva um programa em JS que solicite ao operador dois conjuntos A e B, ambos formados por numeros inteiros, separados por vírgulas. Note que o sistema – por simplicidade – não verifica as digitações. Então, se algo errado ou esquisito, for digitado, o programa deverá ter um comportamento inesperado, errado ou errático.

Depois de ter convertido as entradas em conjuntos (eliminando eventuais elementos duplicados), o programa deverá calcular 4 novos conjuntos:

- União: $A \cup B$
- Intersecção: $A \cap B$
- Diferença: $A - B$
- Produto cartesiano: o conjunto de todos os pares (x_1, x_2) onde $x_1 \in A$ e $x_2 \in B$.

O código usando objetos Set

```
function criarConjunto1(arr) {
  const conjunto = new Set(arr);
  return [...conjunto];
}
function uniao1(conjuntoA, conjuntoB) {
  return [...new Set([...conjuntoA,
        ...conjuntoB])];
}
function intersecao1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       conjuntoB.includes(item));
}
function diferenca1(conjuntoA, conjuntoB) {
  return conjuntoA.filter(item =>
       !conjuntoB.includes(item));
}
```

Código usando apenas listas

```
    function criarConjunto2(arr){
  const conjunto=[]
  for (i=0;i<arr.length;i++){
    for(j=i+1;j<arr.length;j++){
    if (arr[i]==arr[j])
           {arr[j]=-9999;}
 }
  }
  for (i=0;i<arr.length;i++){
    if (arr[i]!=-9999)
         {conjunto.push(arr[i]);}
 }
 return conjunto;
}
function uniao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++)
       {conjunto.push(caa[i]);}
  for (i=0;i<cbb.length;i++){
    if (conjunto.indexOf(cbb[i])==-1)
       {conjunto.push(cbb[i]);}
 }
 return conjunto;
}

function intersecao2(caa,cbb){
  const conjunto=[]
  for (i=0;i<caa.length;i++){
    for (j=0;j<cbb.length;j++){
    if (caa[i]==cbb[j])
           {conjunto.push(caa[i])}
    }
  }
  return conjunto
}
function diferenca2(caa,cbb){
 const conjunto=[]
  for (i=0;i<caa.length;i++){
    if (cbb.indexOf(caa[i])==-1)
           {conjunto.push(caa[i])}
  }
  return conjunto
}

function execu(){
  ca=document.getElementById('ca').value;
  cb=document.getElementById('cb').value;
  ca=ca.split(',')
  cb=cb.split(',')
  ca=ca.map(Number)
  cb=cb.map(Number)
  ca=criarConjunto2(ca)
  cb=criarConjunto2(cb)
  aub=uniao2(ca,cb)
  aib=intersecao2(ca,cb)
  amb=diferenca2(ca,cb)
  pr=' '
  for (i=0;i<ca.length;i++){
    for(j=0;j<cb.length;j++){
   pr=pr+ca[i]+":"+cb[j]+','
 }
  }
  pr= pr.slice(0, -1)  // para tirar a ultima ,
  document.getElementById('aub').value=aub
  document.getElementById('aib').value=aib
  document.getElementById('amb').value=amb
  document.getElementById('apb').value=pr
}
```

☞ Para você fazer

Primeiro implemente este aplicativo em JS. Mostre para o professor.

Avaliação

Agora execute o seu programa com os seguintes conjuntos

$A = \{8, 5, 2, 4, 3\}$
$B = \{3, 1, 4, 9, 5\}$
Responda aqui:

$A \cup B$	
$A \cap B$	
$A - B$	
Produto cartesiano	

508-76182 - ga/ a

A tela A seguir, o visual da tela do programa