_____ / _____ / _____

Programas C com parâmetros

Um programa C em linha de comando precisa ter uma função `main()` definida dentro dele. Essa exigência é para ter a quem passar o controle da execução quando este programa for chamado a partir da linha de comando de uma janela do sistema operacional.

Tipicamente, um programa pode iniciar perguntando coisas ao seu operador, recebendo dele as respostas e a partir daí definindo o que fazer.

Entretanto, existe uma maneira mais rápida de fazer isso. É combinar com o usuário que ele vai fornecer as informações de que o programa precisa junto com o nome do programa. Em outros termos, ao chamar o programa o operador já vai passar a ele os dados que ele precisa. Com isso, há pelo menos duas vantagens:

- evita-se um ciclo de pergunta e resposta
- mas, principalmente o programa deixa de ser interativo, podendo fazer parte de uma sequência de programas postos a rodar em conjunto sem interferência do operador

A esta última característica dá-se o nome de processamento em lote (ou em inglês, *batch processing*).

Para construir programas com este comportamento, a função *main* do seu programa que era escrita como `main()`, passará a ser escrita como `main(int ac, char *av[])`. Você deve escrever a primeira linha de `main` exatamente como acima descrito, sem modificar nada.

O parâmetro `ac` é uma variável inteira que contém o número de parâmetros passados ao seu programa quando ele foi chamado. O menor valor para `ac` é sempre 1, já que o nome do seu programa é considerado neste caso como um parâmetro.

O parâmetro `av[]` é um vetor de caracteres e ele contém o valor dos parâmetros. O primeiro elemento `av[0]` como já se viu contém o nome do seu programa. Os elementos seguintes, desde `av[1]` até `av[n]` contém os valores dos demais parâmetros.

Uma coisa que pode ser feita neste tipo de programa que espera receber (por exemplo) um parâmetro é o seguinte teste:

```
if (ac != 2) {
   printf("Chamada inválida ao programa\n");
   printf("O correto é ...\n");
}
```

Se o parâmetro esperado é um número, o compilador C entrega-o como uma cadeia de caracteres (já que a linha de comando É uma cadeia de caracteres). Antes de usar o parâmetro como uma variável numérica é necessário transformá-lo nisso. Veja como

```
#include<math.h>
int x;
float y;
...
x = atoi(av[1]);
y = atof(av[2]);
...
```

Lembrando que av[1] é o primeiro parâmetro (av[0] é o nome do programa) na forma de uma cadeia de caracteres e x é a variável numérica inteira que contém o valor do parâmetro. y é similar, mas para variáveis de ponto flutuante. Note a diferença entre variáveis inteiras e reais.

☞ Para você fazer

A seguir a definição de alguns programas C que você deve escrever, codificar, compilar, testar, depurar e certificar como corretos. Imprima os códigos fonte C em anexo a esta folha e devolva tudo para o professor. Por favor, identifique claramente onde começa cada programa. Ajude na correção. Obrigado. Se alguma coisa que é pedida ainda lhe é desconhecida, não se avexe. Na biblioteca de livros da UP há centenas de livros sobre C. Na Internet há muito material de referência. O professor está também à sua disposição. A referência que gosto de recomendar é *C Completo e Total* de Herbert Schildt, mas pode usar qualquer outro bom livro de C.

1. Escreva um programa C que receba um parâmetro R real e positivo e imprima a área da superfície esférica da esfera de raio R. Use a fórmula

   $$S = 4 \times \pi \times R^2$$

   Assim, por exemplo se o parâmetro for 11, deve ser impresso 1520.53.

2. Escreva um programa C que receba 3 parâmetros inteiros e positivos. O programa deve imprimir o maior deles. Assim, por exemplo, se os parâmetros forem 10, 12 e 13, o programa deve imprimir 13. Se forem 5, 4 e 5, deve imprimir 5 e se forem 3, 3 e 3 deve imprimir 3.

3. Escreva um programa C que receba um parâmetro R real e positivo e imprima o comprimento da circunferência de raio R. Use a fórmula

   $$C = 2 \times \pi \times R$$

   Assim, por exemplo se o parâmetro for 11, deve ser impresso 69.11.

4. Escreva um programa C que receba um parâmetro N inteiro e positivo na linha de comando e imprima todos os múltiplos de N que são menores do que 100. Assim, por exemplo, se N=49, o programa deve imprimir 49 e 98. Se N=102, o programa não deve imprimir nada e se N=7, deve imprimir 7 14 21 28 35 42 49 56 63 70 77 84 91 98.

5. Imagine um relógio de ponteiros trabalhando na modalidade discreta: Durante 60 segundos inteiros os ponteiros permanecem fixos nas suas posições. Ao virar o minuto, os ponteiros andam ambos para suas novas posições. Como se sabe, os ponteiros têm centro comum, logos eles formam um determinado ângulo entre eles. Escreva um programa C que receba 2 parâmetros: H_1 e M_1, calcule e imprima o menor ângulo em graus formado pelos dois ponteiros. Por exemplo, se os parâmetros forem 12 10, o programa deverá imprimir 55.

6. Escreva um programa C que receba um parâmetro N inteiro e positivo e calcule e imprima o valor S real, assim calculado

   $$S = \frac{2}{1} + \frac{3}{4} + \frac{4}{9} + ... + \frac{n}{n-1^2}$$

   Note que S tem N parcelas. Por exemplo, se $N = 4$ a impressão deve ser 3.506944444 e se $N = 6$ deve ser 3.941388889

7. Escreva um programa C que receba um parâmetro N inteiro e positivo e imprima "PAR"ou "IMPAR"conforme o valor de N. Assim, por exemplo, 6 é par e 7 é ímpar.

8. Escreva um programa C que receba dois parâmetros D e M inteiros e positivos, o primeiro representando um dia e o segundo um mês do ano de 2014. A data é válida, isso não precisa ser testado. O programa deve imprimir quantos dias faltam até 31/dez/2014. Por exemplo, se os números forem 17 5, o programa deve imprimir 258.

9. Escreva um programa C que receba um parâmetro R real e positivo e imprima a área da superfície do círculo de raio R. Use a fórmula

   $$S = \pi \times R^2$$

   Assim, por exemplo se o parâmetro for 11, deve ser impresso 380.13.

10. Escreva um programa C que receba dois parâmetros M e N, inteiros e positivos, ache e imprima a soma dos múltiplos de M que são menores do que N. Então, por exemplo, se $M = 7$ e $N = 70$, o programa deve imprimir 14 21 28 35 42 49 56 63. Note que o próprio M não integra a saída.

11. Sabendo que qualquer primo maior do que 2 pode ser escrito como a diferença de 2 quadrados de números inteiros vizinhos, escreva um programa C que receba N (que é primo maior do que 2, não é preciso testar isto) e imprima os 2 números que quando elevados ao quadrado e subtraídos reproduzem o primo original. Por exemplo, $101 = 51^2 - 50^2$.
    DICA: estude a fórmula $(n+1)^2 - n^2 = 2n + 1 = n + n + 1$

12. Escreva um programa C que receba um parâmetro N inteiro e positivo na linha de comando e imprima todos os múltiplos de 17 que são menores do que 200 e maiores do que N. Assim, por exemplo, se N=49, o programa deve imprimir 51 68 85 102 119 136 153 170 187. Se N=102, o programa deve imprimir 119 136 153 170 187 e se N=7, deve imprimir 17 34 51 68 85 102 119 136 153 170 187.

13. Escreva um programa C que receba 6 parâmetros, a saber: N_1, N_2, N_3, N_4, N_5 e N_6. O programa deve imprimir "DECRESCENTE"se eles estiverem em ordem decrescente e deve imprimir "ESTRITAMENTE DECRESCENTE"se eles estiverem em ordem estritamente decrescente. Uma sequência de números está em ordem crescente se e somente se $N_i \geq N_j \forall i < j$. Uma sequência de números está em ordem estritamente crescente se e somente se $N_i > N_j \forall i < j$. Assim, por exemplo, se a sequencia for 7 7 7 7 7 6, o programa deve imprimir DECRESCENTE. Se for 6 5 4 3 2 1 deve imprimir DECRESCENTE e ESTRITAMENTE DECRESCENTE. Se for 4 5 4 3 2 1 não deve imprimir nada.

Observação importante. Não use o cabeçalho `main()` seguido por `printf()` e `scanf()`. Agindo assim você não estará usando parâmetros, que é uma exigência desta folha. Releia a teoria logo acima, se não entendeu a razão desta observação.