

IF165: Computação Eletrônica: Laboratório

Gurvan Huiban

03 de junho de 2014

Sugerimos usar o ambiente de desenvolvimento Code::Blocks, que deve estar instalado nos computadores do laboratório. Para criar um programa, proceder da forma seguinte:

- Menu `File`, opção `New→File`
- Escolher `C/C++ source` e continuar.
- Quando perguntado, escolher `C`, informar o nome do arquivo e terminar a criação do novo arquivo.
- Não esquecer que o arquivo deve começar com `#include<stdio.h>`.

A compilação e a execução do arquivo compilado são feitas com o menu `Build→Build` e `Build→Run`.

1 Ordenação de vetor

Escreva um programa para ordenar de forma crescente um vetor contendo 10 números inteiros. Para realizar esta tarefa, implemente as funções descritas a seguir.

1.1 Função inicializar

```
void inicializar(int v[10])
```

Preenche o vetor com valores inteiros dados pelo usuário.

1.2 Função imprimir

```
void imprimir(int v[10])
```

Imprime na tela (na mesma linha) o vetor.

1.3 Função encontrarMinimo

```
int encontrarMinimo(int v[10], int k)
```

Procura na parte do vetor definida pelos elementos de índice maior ou igual a k (elementos de índice k , $k+1$, $k+2$ até 9) o elemento de menor valor, e retorna seu índice. O valor k deve ser parâmetro de entrada do subprograma.

Por exemplo, com o seguinte vetor:

Valor	57	96	26	47	19	68	38	37	39	81
Índice	0	1	2	3	4	5	6	7	8	9

Se a função for chamada com $k=5$, a função deve retornar 7, porque 7 é o índice do elemento de menor valor (37) dentre os elementos que tem índice maior ou igual a 6 e menor ou igual a 9.

1.4 Função trocarValor

```
void trocarValor(int v[10], int i, int j)
```

Troca as posições dos elementos de índice i e j no vetor.

1.5 Programa principal

O programa principal deve ler um vetor do usuário e ordená-lo da seguinte forma:

1. Chame a função `inicializar`;
2. Chame a função `imprimir`;
3. Para o contador i , de 0 até 8, execute os seguintes passos:
 - (a) Chame a função `encontrarMinimo`, para pagar o índice do elemento de menor valor dentre os elementos que tem índice maior ou igual a i e menor ou igual a 9. Salve o índice na variável `imin`;
 - (b) Chame a função `trocarValor` para trocar de posição os elementos com índice i e `imin`.
4. Chame a função `imprimir`.

2 Raiz de polinômio

Para encontrar a raiz de um polinômio $p(x) = a_0 + a_1x + \dots + a_nx^n$, ($n > 2$), pode-se aplicar o método de Newton (1), que consiste em refinar uma aproximação inicial x_0 dessa raiz através da expressão:

$$x_{n+1} = x_n - \frac{p(x_n)}{p'(x_n)}$$

$n = 0, 1, 2, \dots$, onde $p'(x)$ é a primeira derivada de $p(x)$.

Usualmente, repete-se esse refinamento até que $|x_{n+1} - x_n| < \epsilon$, $\epsilon > 0$, ou até que m iterações tenham sido executadas.

Para simplificação, vamos considerar que o grau do polinômio p é igual a 4. Os coeficientes do polinômio p são armazenados em um vetor onde o $i^{\text{ésimo}}$ elemento do vetor corresponde ao coeficiente do termo de grau i .

Para solução do problema, deve ser definida:

- Uma função
`float CalcularProximo(float x, float p[5])`
que recebe um ponto x e o vetor dos coeficientes, que calcula e retorna o valor de $x - \frac{p(x)}{p'(x)}$;
- Uma função
`int precisao(float x1, float x2, float e)`
que retorna 1 caso $|x_2 - x_1| < e$, 0 caso contrário;
- O programa principal deve solicitar do usuário o valor dos 5 coeficientes do polinômio p , o valor de epsilon (ϵ), o valor de x_0 , e calcular os termos da sequência (x_n) usando a função `CalcularProximo`, até que $|x_{n+1} - x_n| < \epsilon$, ou até que n seja maior ou igual a 100;
- Imprimir na tela o valor de x encontrado com 5 casas decimais.

Observações:

- Lembre que o valor de x^n pode ser obtido com a função `pow(x, n)`
- Seja um polinômio $p(x) = a_0 + a_1x + \dots + a_nx^n$. Então o polinômio derivado $p'(x) = a_1 + 2a_2x + 3a_3x^2 + \dots + na_nx^{n-1}$.
- Não utilizar variáveis globais para passagem de parâmetros entre as funções.