Universidade Federal De São João Del Rei
Departamento De Ciência Da Computação
Curso De Ciência Da Computação

ROTEIRO 1 - LAB DE PROGRAMACAO II

Adélson de Oliveira Carmo Júnior - 212050019

1 PONTEIROS

1.1 Nesta questão, para testar diferentes valores no vetor, é necessário alterá-lo no código.

Código:

```c
#include <stdio.h>

/* Cabecalho */
int negativos(float *vet, int N);

/* Funcao que confere a quantidade de numeros negativos em um vetor e a retorna*/
int negativos(float *vet, int N){
   int contador = 0;
    /* Percorre o vetor e quando acha um negativo adiciona no contador local*/
   for(int i = 0; i < N; i++){
      if(vet[i] < 0)
         contador++;
   }
    /* Devolve o valor do contador local, que contem a quantidade de numeros negativos*/
   return contador;
}

void main(){
float vetor[5] = {1,-2,-3,4-5};
int tamanho = sizeof(vetor)/sizeof(vetor[0]);
printf("A quantidade de numeros negativos no vetor eh: %d\n",negativos(vetor,tamanho));
}
```

Saída

```
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ gcc questao_1.1.c -o q11
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ ./q11
A quantidade de numeros negativos no vetor eh: 3
```

1.2 Para testar diferentes tamanhos de vetor, é necessário alterar o valor de “tamanho” no código.

Código

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

/* Cabecalho */
float* preencheVetor(float *,int);
float maiorElemento(float *, int);
float menorElemento(float *, int);
float media(float *, int);
void imprimeVetor(float *, int);

/* Funcao que preenche o vetor recursivamente*/
float* preencheVetor(float *vetor, int tamanho){
   srand((time_t)time(NULL));
   if(tamanho!=0)
      vetor = preencheVetor(vetor, tamanho-1);
   vetor[tamanho]= rand() %100;
   return vetor;
}

/* a) Funcao que retorna o maior elemento */
float maiorElemento(float *vet, int N){
   float maior = 0;
   for(int i = 0; i < N; i++){
      if(vet[i] > maior)
         maior = vet[i];
   }
   return maior;
}

/* b) Funcao que retorna o menor elemento */
float menorElemento(float *vet, int N){
   float menor = 10000000000;
   for(int i = 0; i < N; i++){
```

```
    if(vet[i] < menor)
        menor = vet[i];
    }
    return menor;
}

/* c) Funcao que retorna o media entre os elementos do vetor */
float media(float *vet, int N){
    float soma = 0;
    for(int i = 0; i < N; i++)
        soma = soma + vet[i];
    return soma/N;
}

/* Imprime vetor */
void imprimeVetor(float *vetor, int tamanho){
    for(int i = 0; i<tamanho; i++)
    printf("vetor[%d] = %.3f\n", i, vetor[i]);
}



void main(){
    int tamanho = 5;
    float *vetor = (float*)malloc(sizeof(float)*tamanho);

    vetor = preencheVetor(vetor,tamanho);
    imprimeVetor(vetor, tamanho);

    printf("O menor elemento eh: %.3f\n",menorElemento(vetor,5));
    printf("O maior elemento eh: %.3f\n",maiorElemento(vetor,5));
    printf("A media dos elementos eh: %.3f\n",media(vetor,5));
}
```

Saída:

```
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ gcc questao_1.2.c -o q12
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ ./q12
vetor[0] = 74.000
vetor[1] = 20.000
vetor[2] = 12.000
vetor[3] = 74.000
vetor[4] = 15.000
O menor elemento eh: 12.000
O maior elemento eh: 74.000
A media dos elementos eh: 39.000
```

1.3 Caso queira testar diferentes valores para os alunos, é necessário alterá-los no código.
Código

```c
#include <stdio.h>

/* Struct */
typedef struct aluno{
char* nome;
   int matricula;
   int nota;
}Aluno;

/* Cabecalho*/
Aluno criaAluno(Aluno);
void maiorNota(Aluno *, int);
void menorNota(Aluno *, int);

/* Funcao que cria um Aluno */
Aluno criaAluno(Aluno aluno){
   aluno.nome = "";
   aluno.matricula = 0;
   aluno.nota = 0;
   return aluno;
}

/* Funcao que imprime os dados do aluno de maior nota*/
void maiorNota(Aluno *aluno, int tamanho){
   Aluno maior = criaAluno(maior);
   for(int i = 0; i < tamanho; i++){
      if(aluno[i].nota > maior.nota)
         maior = aluno[i];
   }
     printf("O aluno %s, de matricula %d, tem a maior nota: %d\n",
maior.nome, maior.matricula, maior.nota);
}

/* Funcao que imprime os dados do aluno de menor nota*/
void menorNota(Aluno *aluno, int tamanho){
   Aluno menor = criaAluno(menor);
   menor.nota = 1000000;
   for(int i = 0; i < tamanho; i++){
      if(aluno[i].nota < menor.nota)
      menor = aluno[i];
   }
```

```
    printf("O aluno %s, de matricula %d, tem a menor nota: %d\n",
menor.nome, menor.matricula, menor.nota);
}

void main(){
   /* Cria alunos*/
   Aluno aluno1 = criaAluno(aluno1);
   aluno1.nome = "Fulano";
   aluno1.matricula = 1;
   aluno1.nota = 9;

   Aluno aluno2 = criaAluno(aluno1);
   aluno2.nome = "Beltrano";
   aluno2.matricula = 2;
   aluno2.nota = 4;

   Aluno aluno3 = criaAluno(aluno1);
   aluno3.nome = "Ciclano";
   aluno3.matricula = 3;
   aluno3.nota = 10;

   /* Cria um vetor de alunos*/
   Aluno aluno[3] = {aluno1,aluno2, aluno3};

   maiorNota(aluno,3);
   menorNota(aluno,3);
}
```

Saída:

```
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ gcc questao_1.3.c -o q13
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ ./q13
O aluno Ciclano, de matricula 3, tem a maior nota: 10
O aluno Beltrano, de matricula 2, tem a menor nota: 4
```

1.4 Novamente, para testar diferentes valores para A, B e C, é necessário alterá-los no código.
Código:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* Cabecalho */
double delta(float, float, float);
int raizes(float, float, float, float*, float *);
```

```c
/* Funcao que retorna o delta*/
double delta(float A, float B, float C){
   return B*B - (4*A*C);
}

/* Funcao que retorna as raizes caso elas existam*/
int raizes(float A, float B, float C, float* X1, float * X2){
   double delta_local = delta(A,B,C);

   if(delta_local > 0){
      delta_local = sqrt(delta_local);
      *X1 = (float)((-B + delta_local)/2*A);
      *X2 = (float)(-B - delta_local)/2*A;
   return 2;
   }
   else{
      delta_local = sqrt(delta_local);
      if(delta_local == 0){
         *X1 = (float)(-B + delta_local)/2*A;
         *X2 = *X1;
         return 1;
      }
      else
         return 0;
   }
}

void main(){
   float A = 1;
   float B = 4;
   float C = 0;
   float *X1 = (float*)malloc(sizeof(float));
   float *X2 = (float*)malloc(sizeof(float));

   switch (raizes(A,B,C,X1,X2)){
   case 0:
      printf("Nao ha raizes reais, pois o delta eh menor que 0\n");
      break;
   case 1:
       printf("So ha uma raiz real: %f. Isso ocorre devido ao fato de
que o delta eh zero e portanto as raizes sao iguais.\n",*X1);
      break;
```

```
    case 2:
        printf("As duas raizes reais sao: %f e %f\n",*X1,*X2);
        break;
    default:
        printf("Se chegou aqui deu errado XD\n");
        break;
    }
}
```

Saída:

```
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ gcc questao_1.4.c -o q14 -lm
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ ./q14
As duas raizes reais sao: 0.000000 e -4.000000
```

2 RECURSIVIDADE

2.1 Neste caso, o código sempre vai imprimir os valores entre o mínimo e máximo, sendo necessário alterá-los no código para novos testes.

Código:

```
#include <stdio.h>

/* Cabecalho */
void crescente(int, int);
void decrescente(int, int);

/* Funcao que imprime o vetor de forma crescente */
void crescente(int maximo, int minimo){
    if(maximo != minimo)
    crescente(maximo-1, minimo);
    printf("%d ",maximo);
}

/* Funcao que imprime o vetor de forma decrescente */
void decrescente(int maximo, int minimo){
    if(maximo != minimo)
        decrescente(maximo, minimo+1);
    printf("%d ",minimo);
}

void main(){
    int maximo = 5;
```

```
    int minimo = 1;

    printf("Forma crescente: ");
    crescente(maximo, minimo);
    printf("\n");

    printf("Forma decrescente: ");
    decrescente(maximo, minimo);
    printf("\n");
}
```

Saída:

```
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ gcc questao_2.1.c -o q21
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ ./q21
Forma crescente: 1  2  3  4  5
Forma decrescente: 5  4  3  2  1
```

2.2 Para realizar novos testes no vetor, é necessário alterá-lo no código.
Código:

```
#include <stdio.h>

/* Cabecalho */
void ordena(char[], int );
void imprimeVetor(char[], int);


/* Funcao que ordena o vetor de forma descrescente*/
void ordena(char vetor[], int tamanho){
    int posicao_do_maior = 0;
    char aux = 0;
    for(int i = 0; i < tamanho; i++){
        posicao_do_maior = i;

        for (int j= i+1; j < tamanho;j++){
            if(vetor[j] > vetor[posicao_do_maior])
            posicao_do_maior = j;
        }

        if(i != posicao_do_maior){
            aux = vetor[i];
            vetor[i] = vetor[posicao_do_maior];
            vetor[posicao_do_maior] = aux;
        }
```

```
    }
}

/* Funcao que imprime o vetor */
void imprimeVetor(char vetor[], int tamanho){
    if(tamanho != 0)
    imprimeVetor(vetor, tamanho-1);
    printf("%c ", vetor[tamanho-1]);
}

void main(){
    char vetor[5] = {'a','f','r','d','e'};
    printf("Vetor de letras:");
    imprimeVetor(vetor, 5);
    printf("\n");

    ordena(vetor, 5);
    printf("Vetor de letras do maior para o menor:");
    imprimeVetor(vetor, 5);
    printf("\n");
}
```

Saída:

```
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ gcc questao_2.2.c -o q22
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ ./q22
Vetor de letras:  a  f  r  d  e
Vetor de letras do maior para o menor:  r  f  e  d  a
```

2.3 Nesta questão, para testar diferentes valores no vetor, é necessário alterá-lo no código.
Código:

```
#include <stdio.h>

/* Cabecalho */
void ordena(int[], int );
int somaVetor(int[], int);

/* Funcao que ordena o vetor de forma crescente*/
void ordena(int vetor[], int tamanho){
    int posicao_do_menor = 0;
    int aux = 0;
    for(int i = 0; i < tamanho; i++){
        posicao_do_menor = i;
```

```
      for (int j= i+1; j < tamanho;j++){
         if(vetor[j] < vetor[posicao_do_menor])
            posicao_do_menor = j;
      }

      if(i != posicao_do_menor){
         aux = vetor[i];
         vetor[i] = vetor[posicao_do_menor];
         vetor[posicao_do_menor] = aux;
      }

   }
}

/* Funcao que soma o vetor */
int somaVetor(int vetor[], int posicao){
   int soma = vetor[posicao];
   if(posicao != 0)
   soma = soma + somaVetor(vetor, posicao-1);
   return soma;
}

void main(){
int vetor[5] = {5,4,3,2,1};

   ordena(vetor, 5);

   printf("Soma do vetor: %d\n", somaVetor(vetor, 4));
}
```

Saída:

```
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ gcc questao_2.3.c -o q23
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ ./q23
Soma do vetor: 15
```

2.4 Caso queira testar outros valores para “n” e “m”, basta alterá-los no código.
Código:

```
#include <stdio.h>

/* Cabecalho */
int multiplica(int, int);

/* Funcao que multiplica um valor m, n vezes */
```

```c
int multiplica(int n, int m){
   int multiplicacao = m;
   if(n!=1)
      multiplicacao = multiplicacao * multiplica(n-1, m);
   return multiplicacao;
}

void main(){
   int m = 2;
   int n = 5;

   printf("Soma do vetor: %d\n", multiplica(n,m));
}
```

Saída:

```
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ gcc questao_2.4.c -o q24
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ ./q24
Soma do vetor: 32
```

2.5 Devido ao fato que a função rand é usada nessa questão, a única mudança que pode ser feita no código, para novos testes, é no tamanho do vetor. Para isso, basta alterar a variável tamanho.

Código:

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

/* Cabecalho */
int* preencheVetor(int *, int );
void ordena(int*, int );
void imprimeCrescente(int *, int , int );
void imprimeDecrescente(int *, int );

/* a) Funcao que preenche o vetor recursivamente*/
int* preencheVetor(int *vetor, int tamanho){
   srand((time_t)time(NULL));
   if(tamanho!=0)
      vetor = preencheVetor(vetor, tamanho-1);
   vetor[tamanho]= rand() %100;

   return vetor;
}
```

```c
/* Funcao que ordena o vetor de forma descrescente*/
void ordena(int *vetor, int tamanho){
   int posicao_do_maior = 0;
   int aux = 0;
   for(int i = 0; i < tamanho; i++){
      posicao_do_maior = i;

      for (int j= i+1; j < tamanho;j++){
         if(vetor[j] > vetor[posicao_do_maior])
            posicao_do_maior = j;
      }

      if(i != posicao_do_maior){
         aux = vetor[i];
         vetor[i] = vetor[posicao_do_maior];
         vetor[posicao_do_maior] = aux;
      }

   }
}


/* b) Funcao que imprime o vetor */
void imprimeCrescente(int *vetor, int tamanho, int posicao){
   if(posicao != tamanho-1)
   imprimeCrescente(vetor, tamanho, posicao+1);
   printf("%d ", vetor[posicao]);
}

/* b) Funcao que imprime o vetor */
void imprimeDecrescente(int *vetor, int tamanho){
   if(tamanho != 1)
   imprimeDecrescente(vetor, tamanho-1);
   printf("%d ", vetor[tamanho-1]);
}


/* c) Funcao que busca o maior elemento no vetor e retorna-o*/
int devolveMaior(int *vetor, int tamanho, int maior){
   if(tamanho != 0)
      maior = devolveMaior(vetor, tamanho-1, maior);
   if(vetor[tamanho]> maior)
      maior = vetor[tamanho];
```

```
    return maior;
}



void main(){
int tamanho = 5;
    int *vetor = (int*)malloc(sizeof(int)*tamanho);
    vetor = preencheVetor(vetor, tamanho);
    printf("Vetor: ");
    imprimeDecrescente(vetor, tamanho);
    printf("\n");

    ordena(vetor, tamanho);
    printf("Vetor crescente:");
    imprimeCrescente(vetor, tamanho, 0);
    printf("\nVetor decrescente:");
    imprimeDecrescente(vetor, tamanho);
       printf("\nMaior elemento do vetor: %d\n", devolveMaior(vetor,
tamanho, 0));
}
```

Saída:

```
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ gcc questao_2.5.c -o q25
adelson@adelson-junior:~/UFSJ/6_Periodo/lab_AEDSII/roteiro1$ ./q25
Vetor: 29  61  79  6  22
Vetor crescente:6  22  29  61  79
Vetor decrescente:79  61  29  22  6
Maior elemento do vetor: 79
```

2.6

O algoritmo foi devidamente analisado e entendido, com o auxílio das diferentes entradas.