

## **GABARITO DA LISTA DE EXERCÍCIOS: 2ª Unidade**

1) Faça um programa que leia um valor N inteiro e calcule por meio de uma função o seu dobro, após isso, imprima o número e o seu dobro na tela.

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

int dobro(int x)
{
    x = 2*x;
    return x ;
}

int main()
{
    int n;
    printf("Digite um numero\n");
    scanf ("%d",&n);
    n = dobro(n);
    printf("O dobro : %d\n",n);
    return 0 ;
}
```

2) Faça um programa que receba dois valores inteiros X e N e calcule por meio de uma função o valor de X elevado a N, após isso imprima o resultado na tela.

```
#include <stdio.h>
#include <stdlib.h>
int potencia(int x, int n)
{
    int i = 1 , num ;
    num = x ;
    if ( n == 0 )
    {
        num = 1 ;
    }
    while( i < n )
```

```

    {
        num = num*x ;
        i++;
    }
    return num ;
}
int main()
{
    int base,expoente,resu;
    printf("Digite a base\n");
    scanf("%d",&base);
    printf("Digite o expoente\n");
    scanf("%d",&expoente);
    resu = potencia(base,expoente);
    printf("%d elevado a %d = %d",base,expoente,resu);
    return 0 ;
}

```

3) Faça um programa que leia um valor n inteiro e calcule por meio de uma função o seu dobro, após isso, imprima o resultado na tela.

```

#include <stdio.h>
#include <stdlib.h>
int dobro(int x)
{
    x = 2*x;
    return x;
}
int main()
{
    int n;
    printf("Informe um valor para n:\n");
    scanf("%d",&n);
    n = dobro(n);
    printf("O seu dobro e %d\n",n);
    return 0;
}

```

4) Faça um programa que leia um valor N inteiro e, através de uma função, atualize seu valor para N+1 e imprima na tela seu resultado.

```

#include <stdio.h>
#include <stdlib.h>
void atualiza(int x)

```

```

{
    x = x + 1;
    printf("O novo valor de n e %d\n",x);
}
int main()
{
    int n;
    printf("Informe um valor para n:\n");
    scanf("%d",&n);
    atualiza(n);
    return 0;
}

```

5) Faça um programa que leia um vetor de inteiros X de 3 posições por meio de uma função chamado LeVetor, após sua leitura, imprima o vetor na tela.

```

#include <stdio.h>
#include <stdlib.h>
void LeVetor()
{
    int vetor[3],i;
    printf("Digite os elementos do vetor:\n");
    for(i = 0; i < 3; i++)
    {
        printf("Elemento %d:\n",i+1);
        scanf("%d",&vetor[i]);
    }
    printf("Vetor:\n[ ");
    for(i = 0; i < 3; i++)
    {
        printf("%d ",vetor[i]);
    }
    printf("]\n");
}
int main()
{
    LeVetor();
    return 0;
}

```

6) Faça um programa em C que:

- a. Crie a função FAT que recebe um número inteiro e retorna o fatorial do número.
- b. Lê do teclado um número inteiro N. Se  $N < 0$ , exiba uma mensagem de erro e peça novamente. O programa só prossegue se for digitado  $N \geq 0$ .

c. Imprime na tela o fatorial de N e o fatorial de N+4.

OBS. Definição matemática de fatorial:  $\text{fat}(0) = 1$  e  $\text{fat}(n) = n * \text{fat}(n-1)$ .

Exemplo.  $\text{fat}(5) = 5 * 4 * 3 * 2 * 1$ .

```
#include <stdio.h>
#include <stdlib.h>
int FAT(int x)
{
    int f = 1,i;
    for(i=x; i>0; i--)
    {
        f = f*i;
    }
    return f;
}
int main()
{
    int n,fat;
    do
    {
        printf("Digite um numero:\n");
        scanf("%d",&n);
    }
    while (n < 0);
    fat = FAT(n);
    printf("Fatorial de N = %d\n",fat);
    printf("Fatorial de N + 4 = %d\n",FAT(n+4));
    return 0;
}
```

## Vetores e Matrizes

7) Faça um programa que leia um vetor com 5 nomes, e imprima estes nomes enumerados pela ordem em que foram digitados.

Exemplo de saída: 1- José

2-Carlos

3-Marta

4-Lucas

5-Alex

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

```

#include <stdlib.h>
int main( )
{
    char nomes[5][21];
    int i;
    for (i = 0; i < 5; i++)
    {
        printf("Digite o nome %d: ", i+1);
        scanf(" %s", nomes[i]);
    }
    printf("\n");
    for (i = 0; i < 5; i++)
    {
        printf("%d- %s\n", i+1, nomes[i]);
    }
    return 0;
}

```

8) Faça um programa em C para ler um vetor de 10 números inteiros, calcule e imprima a média aritmética de seus elementos., em seguida atualize seu valor com o dobro do valor anterior, e imprima o vetor atualizado.

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int vetor[10];
    int i;
    float soma = 0;
    for (i = 0; i < 10; i++)
    {
        printf("Digite o valor da posicao %d do vetor: ", i+1);
        scanf("%d", &vetor[i]);
        soma = soma + vetor[i];
        vetor[i] = vetor[i] * 2;
    }
    printf("\nO valor da media aritmetica dos elementos do vetor eh:
%.2f\n", soma/10);
    printf("\nVetor atualizado com o dobro do seu valor:\n");
    for (i = 0; i < 10; i++)
    {
        printf("%d ", vetor[i]);
    }
    printf("\n");
    return 0;
}

```

```
}
```

9) Faça um programa para ler 2 vetores A e B com 5 elementos reais e construir um terceiro vetor S obtido com os valores comuns a A e B. Ao final, imprimir os 3 vetores.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    float A[5], B[5], S[5];
    int i, j, k = 0;
    for (i = 0; i < 5; i++)
    {
        printf("Digite o valor de A[%d] e de B[%d]: ", i+1, i+1);
        scanf("%f %f", &A[i], &B[i]);
    }
    for (i = 0; i < 5; i++)
    {
        for (j = 0; j < 5; j++)
        {
            if (A[i] == B[j])
            {
                S[k] = A[i];
                k++;
            }
        }
    }
    printf("\nVetor A: ");
    for (i = 0; i < 5; i++) {
        printf("%.2f ", A[i]);
    }
    printf("\nVetor B: ");
    for (i = 0; i < 5; i++) {
        printf("%.2f ", B[i]);
    }
    printf("\nVetor S: ");
    for (i = 0; i < k; i++)
    {
        printf("%.2f ", S[i]);
    }
    printf("\n");
    return 0;
}
```

10) Uma vendedora de frutas quer analisar como foi sua venda no dia. Supondo que ela venda por gramas, faça um programa em C que:

- a) Leia um número N correspondente ao número de vendas do dia ( $10 \leq N \leq 1.000$ ).
- b) Leia um vetor de números reais chamado Peso, correspondente ao peso de uma das vendas.

Calcule e imprima:

- a) O peso médio das vendas.
- b) O maior e o menor peso vendidos.
- c) Supondo que 1 kg custe R\$ 4.35, imprima quando foi arrecadado no dia.

```
#include <stdio.h>
int main ()
{
    int n, i, aux;
    float soma, maior, menor, auxr;
    float peso[1000];
    do {
        printf ("Digite o numero de vendas (entre 10 e 1000):");
        scanf ("%d", &n);
    } while (n<10 || n>1000);
    for (i=0; i<n; i++) {
        aux = i+1;
        printf ("Digite o valor do peso da venda %d\n", aux);
        scanf ("%f", &peso[i]);
        if (i==0) {
            maior = peso[i];menor = peso[i];
            soma = peso[i];
        }
        if (i>0) {
            maior = maior > peso[i] ? maior :peso[i];
            menor = menor < peso[i] ? menor :peso[i];
            soma += peso[i];
        }
    }
    auxr = (soma/n);
    printf ("O peso medio das vendas e: %f", auxr);
    printf ("O maior peso vendido foi: %f", maior);
    printf ("O menor peso vendido foi: %f", menor);
    auxr = (soma/1000)*4.35;
    printf ("A arrecadação total foi de R$ %f", auxr);
    return 0;
}
```

- 11) Faça um programa que:
- a) Leia uma matriz 3x3 de números inteiros.
  - b) Imprima-a em formato matricial.
  - c) Leia um número inteiro k.
  - d) Atualize a matriz com seu valor multiplicado por k, e imprima-a no formato matricial.

```
#include <stdio.h>
int main ()
{
    int i, j, k;
    int mat [3][3];
    for (i=0; i<3; i++) {
        for (j=0; j<3; j++) {
            printf ("Digite o valor da posicao %d, %d: ", i+1, j+1);
            scanf ("%d", &mat[i][j]);
        }
    }
    for (i=0; i<3; i++) {
        for (j=0; j<3; j++){
            printf ("%d", mat[i][j]);
            printf (" ");
        }
        printf ("\n");
    }
    printf ("Digite um valor k para multiplicar a matriz: ");
    scanf ("%d", &k);
    printf ("O valor da matriz atualizada e:\n");
    for (i=0; i<3; i++) {
        for (j=0; j<3; j++) {
            mat[i][j] *= k;
            printf ("%d", mat[i][j]);
            printf (" ");
        }
        printf ("\n");
    }
    return 0;
}
```

- 12) Faça um programa que:
- a) Leia uma número N inteiro entre 3 e 10.
  - b) Leia uma matriz de números reais de tamanho N x N. Obs.: Caso exista algum número menor que zero nesta matriz, atribua a este termo o valor zero automaticamente.
  - c) Calcule e imprima a soma da diagonal principal da matriz lida, e a mesma no formato matricial.



```

#include <stdio.h>
int main ()
{
    float mat[10][10];
    float somadiagonal;
    int i, j, n;
    do {
        printf ("N=?");
        scanf ("%d", &n);
    } while (n<3 || n>10);
    somadiagonal = 0;
    for (i=0; i<n; i++) {
        for (j=0; j<n; j++) {
            printf ("Digite o valor da posição %d,%d da matriz: ",
i+1, j+1);

            scanf ("%f", &mat[i][j]);
            if (mat[i][j]<0)
                mat[i][j] = 0;
            if (i==j)
                somadiagonal += mat[i][j];
        }
    }
    printf ("A soma da diagonal principal da matriz e: %f\n",
somadiagonal);
    for (i=0; i<n; i++) {
        for (j=0; j<n; j++) {
            printf ("%f", mat[i][j]);
            printf (" ");
        }
        printf ("\n");
    }
    return 0;
}

```

13) Faça um programa que calcula o valor patrimonial dos produtos de uma empresa. O programa deve ler 3 vetores com 5 elementos cada: COD (números inteiros), QUANT (números inteiros) e VAL (números reais). Correspondentes ao código do produto, quantidade em estoque e valor unitário. O programa deve fazer a leitura dos valores referentes a cada vetor e calcular o valor patrimonial dos produtos, fazendo somatório da multiplicação da quantidade em estoque pelo valor unitário, utilizando os dados armazenados nos vetores QUANT e VAL. Imprima o Código, seguido do valor patrimonial do produto.

```

#include <stdio.h>
int main ()
{
    int cod[5], quant[5];
    float valor[5], vpat[5];
    int i;
    for (i=0; i<5; i++) {
        printf ("Digite o codigo, a quantidade e o valor do produto
%d\n", i+1);
        scanf ("%d%d%f", &cod[i], &quant[i], &valor[i]);
        vpat[i]=(quant [i]*valor[i]);
    }
    for (i=0; i<5; i++) {
        printf ("codigo: %d, valor patrimonial: %f\n", cod[i],
vpat[i]);
    }
    return 0;
}

```

14) Faça uma tabela que enumere as distâncias entre várias cidades, ou seja, uma matriz simétrica: os termos em relação à diagonal principal desta matriz são iguais ( $Mat[i,j] = Mat[j,i]$ ). Obviamente a digitação de uma matriz com esta propriedade pode ser simplificada, devendo-se digitar apenas os termos que estão acima da diagonal principal. Pede-se um programa em C para:

- a) Ler um vetor com o nome de 5 cidades.
- b) Preencher uma matriz simétrica com a distância entre as cidades.
- c) Exibir na tela o conteúdo desta matriz.

```

#include <stdio.h>
int main ()
{
    char cidades[5][100];
    float distancia[5][5];
    int i, j;
    for (i=0; i<5; i++){
        printf ("Informe o nome da cidade %d", i+1);
        gets (cidades[i]);
    }
    for (i=0; i<5; i++) {
        for (j=0; j<5; j++) {
            if (i==j)
                distancia[i][j] = 0;
            if (j>i){

```

```

        printf ("Digite a distancia de ");
        puts (cidades[i]);
        printf (" para ");
        puts (cidades[j]);
        scanf ("%f", &distancia[i][j]);
        distancia[j][i] = distancia[i][j];
    }
}
}
for (i=0; i<5; i++) {
    puts (cidades[i]);
    for (j=0; j<5; j++) {
        printf ("%f ", distancia[i][j]);
    }
    printf ("\n");
}
return 0;
}

```

15) Uma conta corrente bancária é composta por dois códigos: Código de Agência, com 4 dígitos, e código de conta, com 6 dígitos. Além de um dígito verificador para a agência e outro para a conta. Para se certificar de que os códigos foram digitados corretamente, é feita a comparação do dígito verificador (DV) por um outro dígito calculado (DC). O cálculo do DC é feito nos algarismos do código correspondente multiplicando-se os algarismos do código pelos números primos (1,2,3,5,7,11) e o resto da divisão da soma por 9, como mostrado abaixo:

- Código da conta digitado = 250126, dígito verificador digitado (DV) = 7
- Soma dos produtos dos algarismos do código da conta pelos números primos:  $SP = 2*1 + 5*2 + 0*3 + 1*5 + 2*7 + 6*11 = 97$
- Dígito verificador calculado (DC = resto da divisão de SP por 9) = 7
- Como o dígito verificador DV é igual a DC, conclui-se que a conta foi digitada corretamente. Para o código da agência (4 dígitos), utilizam-se apenas os primos 1,2,3 e 5.

Fazer um programa C que utilize 3 vetores: Um para o código da agência (CA); Outro para o código da conta (CC); Outro para os números primos (NP), conforme os passos abaixo:

- a) Construir o vetor NP com números primos 1,2,3,5,7 e 11;
- b) Ler os algarismos do código da agência para o vetor CA e o dígito verificador DV do código da agência;
- c) Calcular o dígito verificador calculado, DC, utilizando os 4 primeiros elementos de NP;
- d) Se  $DV \neq DC$ , emitir mensagem de erro e encerrar o programa;
- e) Ler os algarismos do código da conta para o vetor CC e o dígito verificador DV do código da conta;

- f) Calcular o dígito verificador calculado, DC;
- g) Se DV != DC, emitir mensagem de erro e encerrar o programa;
- h) Se os códigos da agência e da conta foram digitados corretamente, imprimir os mesmos e a mensagem "Foram digitados corretamente".

```
#include <stdio.h>
int main () {
    int ca[4];
    int cc[6];
    int np[6];
    int i, dv, dc, sp;
    np[0] = 1;
    np[1] = 2;
    np[2] = 3;
    np[3] = 5;
    np[4] = 7;
    np[5] = 11;
    sp = 0;
    for (i=0; i<4; i++) {
        printf ("Digite o algarismo %d do codigo da agencia: ", i+1);
        scanf ("%d",& ca[i]);
        sp += (ca[i]*np[i]);
    }
    printf ("Digite o digito verificador do codigo de agencia:");
    scanf ("%d", &dv);
    dc = sp%9;
    if (dc==dv) {
        sp = 0;
        for (i=0; i<6; i++) {
            printf ("Digite o algarismo %d do codigo da conta: ", i+1);
            scanf ("%d", &cc[i]);
            sp += (cc[i]*np[i]);
        }
        dc = sp%9;
        printf ("Digite o digito verificador do codigo de conta:");
        scanf ("%d", &dv);
        if (dc==dv)
            printf ("Foram digitados corretamente.");
        else
            printf ("Codigo de conta e verificador incompativeis.");
    } else
        printf ("Codigo de agencia e digito verificador incompativeis.");
    return 0;
}
```

16) Faça um programa em C que identifica os corredores classificados para os jogos olímpicos. O programa deve ler do teclado:

- a) Receber o número  $t$  de corredores participantes, no intervalo de  $[2, 20]$ .
- b) O nome de cada atleta deverá ser lido do teclado ser armazenado num vetor.
- c) O valor para a variável real limite, também deverá ser lido do teclado. Esta variável corresponde ao tempo máximo permitido para alcançar a classificação para as olimpíadas (ex. 7s e 20ms).
- d) Cada atleta terá direito a três tomadas de tempo. Armazenar os tempos dos corredores numa matriz  $t \times 3$ , na qual cada linha identifica um atleta e cada coluna seu respectivo tempo.

O programa deverá apresentar na tela:

- a) O nome e menor tempo de cada atleta classificado para as olimpíadas.  
Ex.: Lista de Classificados – Tempo limite 14 segundos.
  - João com o tempo de 12 segundos e 0 milésimos de segundo.
  - Fernando com o tempo de 13 segundos e 450 milésimos de segundo.
- b) A mensagem “Não há nenhum atleta classificado.” Deverá ser exibida caso nenhum atleta atinja o índice olímpico.

```
#include <stdio.h>
int main () {
    int t, i, j, k, cont, seg, mili;
    float limite, menor;
    float m[20][3];
    char nome[20][60];
    do {
        printf ("Informe o numero de participantes (entre 2 e 20): ");
        scanf ("%d", &t);
    } while (t<2 || t>20);
    for (i=0; i<t; i++) {
        printf ("Informe o nome do participante %d:", i+1);
        gets (nome[i]);
    }
    printf ("Informe o tempo maximo para a classificacao para os jogos olimpicos: ");
    scanf ("%f", &limite);
    for (i=0; i<t; i++) {
        for (j=0; j<3; j++) {
            printf ("%s . tempo %d\n", nome[i], j+1);
            scanf ("%f", &m[i][j]);
        }
    }
}
```

```

    cont =0;
    printf ("Lista de classificados - tempo limite: %f segundos",
limite);
    for (i=0; i<t; i++) {
        for (j=0; j<3; j++) {
            {
                if (m[i][j]<limite) {
                    cont+= 1;
                    menor = m[i][j];
                    for (k=1; k<3; k++) {
                        if (menor>m[i][k])
                            menor = m[i][k];
                    }
                    seg = menor;
                    mili = (menor-seg)*100;
                    printf ("%s com o tempo de %d segundos e %d milésimos de
segundo.", nome[i], seg, mili);
                }
            }
        }
    }
    if (cont==0)
        printf ("Não há nenhum classificado.");
    return 0;
}

```

17) Na primeira fase de um concurso público foram realizadas provas escritas de português, matemática e conhecimentos gerais. Fazer um programa em C para:

- a) Ler um inteiro N no intervalo  $10 < N \leq 1000$ ;
- b) Em processo repetitivo: Ler nome e notas das provas de português, matemática e conhecimentos gerais dos N candidatos, colocando seus valores nos vetores NOME, NOTAP, NOTAM e NOTACG, respectivamente. Considerando os pesos 2, 3 e 1 para as provas de português, matemática e conhecimentos gerais, respectivamente, calcular a média ponderada pelos pesos para todos os N candidatos, armazenando nos vetores CLAS e MEDIA, os nomes e média ponderada dos candidatos classificados, ou seja, que conseguiram média  $\geq 7,0$ .
- c) Calcular a média geral dos candidatos classificados (média aritmética das médias dos candidatos classificados).
- d) Imprimir a relação de todos os candidatos classificados e que conseguiram média maior ou igual à média geral (calculada no item c).

Obs.:

- i. O cálculo da média ponderada pode ser realizado utilizando a seguinte fórmula:

$$m = (x_1p_1 + x_2p_2 + \dots + x_np_n)/(p_1 + p_2 + \dots + p_n),$$

onde x corresponde à nota e p ao peso associado.

- ii. Fazer a validação do valor de N até que seja digitado um valor válido.
- iii. Se não houver nenhum candidato classificado, imprimir mensagem indicativa e não processar os itens c) e d).

```
#include <stdio.h>
int main () {
    int n, i, j;
    float med, s;
    float notap[1000], notam[1000], notacg[1000], media[1000];
    char nome[1000][80], clas[1000][80];
    do {
        printf ("Digite um valor para n: ");
        scanf ("%d", &n);
    } while (n<10 || n>1000);
    j =0;
    for (i=0; i<n; i++) {
        printf ("Digite o nome do candidato %d", i+1);
        scanf ("%s", nome[i]);
        printf ("Digite as notas de portugues, matematica e conhecimentos
gerais desse aluno,
                respectivamente: ");
        scanf ("%f%f%f", &notap[i], &notam[i], &notacg[i]);
        med = ((2*notap[i])+(3*notam[i])+notacg[i])/6;
        if (med>=7) {
            j = j+1;
            clas[j] = nome[i];
            media[j]=med;
        }
    }
    if (j==0)
        printf ("Não houve nenhum classificado.");
    else {
        s=0;
        for (i=0; i<j; i++)
            s += media[i];
        s /= j;
        printf ("Media dos candidatos classificados: %f", s);
        for (i=0; i<j; i++) {
            if (media[i]>s)
                printf ("%s, Media: %f", nome[i], media[i]);
        }
    }
    return 0;
```

}

18) Faça um programa em C que calcule a distância entre diferentes cidades, satisfazendo as restrições abaixo:

O programa deve ler do teclado:

- Um vetor de caracteres Cidades que armazene o nome de m diferentes cidades. O valor de m deve ser fornecido pelo usuário – validar para que o valor de m esteja no intervalo [2,10];
- Uma matriz Distancias m x m correspondente a distância entre as m cidades definidas no vetor Cidades. Obs.: Nesta matriz não o valor da diagonal principal (representado pela distância de uma cidade para ela mesma) deve ser atribuído automaticamente como 0, ou seja, não deve ser permitida a digitação. Não é necessário supor que a distância entre a cidade A e B deverá ser igual à distância entre B e A. Por exemplo, na tabela abaixo a distância entre A e B é 90 Km, e a distância entre B e A é de 85 Km.
- O valor para a variável Consumo, correspondendo ao consumo de combustível médio do veículo utilizado, ou seja, quantos quilômetros este veículo percorre com um litro de combustível.

O programa deverá calcular e apresentar na tela:

- Todos os percursos entre as cidades que não ultrapassem 250 Km (os percursos são compostos pelos nomes das cidades de origem e pelos nomes das cidades de destino).
- Todos os percursos (nome e cidade de origem e nome da cidade de destino), juntamente com a quantidade de combustível necessária para o veículo percorrê-los.

	A	B	C	D	E
A	0	90	280	160	45
B	85	0	359	251	80
C	340	403	0	102	135
D	154	256	99	0	50
E	39	76	130	48	0

### Exemplo de saída

Supondo a matriz Distancias acima, o vetor Cidades com os valores A, B, C, D, e E, o Consumo com valor igual a 10, a saída apresentada deve ser (foram apresentados somente dois exemplos de saída solicitados em “a” e “b”):

Distancia: 85 entre A e B.

Distancia: 160 entre A e B.

...

Consumo entre A e B = 8.50

Consumo entre A e C = 28.00

```
#include <stdio.h>
int main () {
    int i, j, k, m;
```



```

float consumo, c;
float distancias[10][10]
char cidades[10][60];
do {
    printf ("Digite a quantidade de cidades: ");
    scanf ("%d", &m);
} while (m<2 || m>10);
for (i=0; i<m; i++) {
    printf ("Digite o nome da cidade %d", i+1);
    scanf ("%s", cidades [i]);
}
for (i=0; i<m; i++) {
    for (j=0; j<m; j++) {
        if (i==j)distancias[i][j] = 0;
        else {
            printf ("Informe a distancia entre %s e %s: ",
cidades[i], cidades[j]);
            scanf ("%f", &distancias[i][j]);
        }
    }
}
printf ("Digite um valor para o consumo medio do veiculo: ");
scanf ("%f", &consumo);
printf ("Distancias menores ou iguais a 250km: ");
for (i=0; i<m; i++) {
    for (j=0; j<m; j++) {
        if (i!=j && distancia[i][j]<=250)
            printf ("De %s para %s", cidades[i], cidades[j]);
    }
}
for (i=0; i<m; i++) {
    for (j=0; j<m; j++) {
        if (i!=j) {
            c = (distancias[i][j]/consumo);
            printf ("Distancia: %f km entre %s e %s", distancias[i]
[j], cidades[i], cidades[j]);
            printf ("Consumo entre %s e %s: %f L", cidades[i],
cidades[j], c);
        }
    }
}
return 0;
}

```

19) O Comitê da Copa do Mundo 2014 cadastrou todos os jogadores com dados referentes aos atletas que vão participar da copa, incluindo: país que representa, o nome e a altura dos jogadores. Fazer um programa em C para executar os seguintes passos:

a) Ler os dados referentes aos jogadores conforme especificação abaixo, armazenando-os nos vetores PAIS, NOME e ALTURA. País do atleta, com 3 caracteres literais; Nome do atleta, com 15 caracteres literais e Altura do atleta, com valor real;

OBS.: 1 – Considerar um máximo de 800 atletas na competição;

2 – Os dados devem ser digitados via teclado, atleta por atleta, até que seja digitado 'ZZZ' para o código do país (este dado não pertence a nenhum atleta e serve apenas para indicar o final da entrada de dados).

b) Após a digitação dos dados, calcular a altura média dos atletas cadastrados, imprimindo, na tela, o valor calculado;

c) Listar, na tela, a relação dos jogadores do Brasil (código do país 'BRA') que tenham altura maior ou igual à média geral calculada anteriormente;

```
#include <stdio.h>
int main () {
    int i, j;
    float altura, media;
    float alt[800];
    char nome[800][16], pais[800][4];
    printf ("Digite XXX");
    scanf ("%s", pais[0]);
    i=0;
    altura=0;
    while (pais[i] != 'ZZZ') {
        printf ("Entre o pais: ");
        scanf ("%s", pais[i]);
        if (pais[i] != 'ZZZ') {
            printf ("Entre o nome: ");
            scanf ("%s", nome[i]);
            printf ("Entre a altura: ");
            scanf ("%f", &alt[i]);
            altura += alt[i];
            i +=1;
        }
    }
    media = (altura/i);
    j=1;
    while (j<=i) {
        if (alt[j]>media && pais[j] = 'BRA')
            printf ("%s", nome[j]);
        j += 1;
    }
}
```

```

    }
    return 0;
}

```

20) Faça um programa em C que:

- Crie uma função DOBRAVETOR que recebe 2 parâmetros: 1 vetor de inteiros e um número inteiro representando o tamanho do vetor. A função multiplica cada elemento do vetor por 2;
- Le do teclado um vetor de inteiros de tamanho 4;
- Imprime na tela os elementos do vetor multiplicados por 2 e multiplicados por 4.

```

#include <stdio.h>
#include <stdlib.h>
void dobra(int v[], int n) {
    int i;
    for(i=0; i<n; i++) {
        v[i] = v[i]*2;
    }
}
int main() {
    int vetor[4],i;
    printf("Digite os 4 elementos do vetor:\n");
    for(i=0; i<4; i++) {
        printf("Elemento da posicao %d:\n",i);
        scanf(" %d",&vetor[i]);
    }
    dobra(vetor,4);
    for(i=0; i<4; i++) {
        printf("%d ",vetor[i]);
    }
    printf("\n");
    dobra(vetor,4);
    for(i=0; i<4; i++) {
        printf("%d ",vetor[i]);
    }
    system("PAUSE");
    return 0;
}

```

21) O índice de massa corpórea (IMC) é calculado através da fórmula (peso/Altura<sup>2</sup>) e, segundo critério da Organização Mundial de Saúde, classificado por categoria, a saber: IMC < 18.5(abaixo do peso);  $18.5 \leq \text{IMC} < 25$  (peso normal);  $\text{IMC} \geq 25$  (acima do peso). Fazer um programa C para:

- Ler um inteiro N ( $5 \leq N \leq 80$ ) correspondente ao número de atletas de uma determinada competição desportiva. Obs.: o programa deve ficar lendo N até ser digitado um N válido.

- b) Ler, para os vetores de elementos reais ALT e PESO, as alturas e os pesos dos N atletas;
- c) Com base nos vetores lidos e utilizando uma função, calcular o IMC de cada participante e o número de atletas pertencentes a cada categoria “abaixo do peso”, “peso normal” ou “acima do peso”, em conformidade com os critérios da Organização Mundial de Saúde, descritos acima. A função deve ter como parâmetros os vetores ALT e PESO, o número de atletas (N), e 3 inteiros (NP1, NP2 e NP3) correspondentes ao número de atletas de cada categoria. Obs: NP1, NP2 e NP3 armazenarão o número de participantes da categoria 1 (abaixo do peso), da categoria 2 (peso normal) e da categoria 3 (acima do peso), que devem ser passados por referência.
- d) Imprimir os valores de NP1, NP2 e NP3 como mostrado ao lado;  
*Obs.: Tais valores não devem ser impressos dentro da função;*  
*Obs.: A função não deve fazer uso de variáveis globais;*

Obs.: Essa questão requer o uso de ponteiros, assunto da unidade 3.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
void categ(float *al,float *pe,int na, int *n1,int *n2,int *n3) {
    int i;
    float imc;
    *(n1)=0;
    *(n2)=0;
    *(n3)=0;
    for(i=0; i<na; i++) {
        imc=pe[i]/(al[i]*al[i]);
        if (imc < 18.5) {
            *(n1)=*(n1)+1;
        } else if(imc < 25.0) {
            *(n2)=*(n2)+1;
        } else *(n3)=*(n3)+1;
    }
}
int main() {
    float alt[80], peso[80];
    int i,np1,np2,np3,n;
    do {
        printf("Digite O numero de atletas:\n");
        scanf(" %d",&n);
    } while ((n < 5) || (n > 80));
    for(i=0; i<n; i++) {
        printf("Altura do atleta %d:\n",i+1);
        scanf(" %f",&alt[i]);
        printf("Peso do atleta %d:\n",i+1);
        scanf(" %f",&peso[i]);
    }
```

```
}  
categ(alt,peso,n,&np1,&np2,&np3);  
printf("Atletas abaixo do peso: %d\n",np1);  
printf("Atletas com peso normal: %d\n",np2);  
printf("Acima do peso: %d\n",np3);  
system("PAUSE");  
return 0;  
}
```