

Resolução da ficha formativa Nº 13

1- Digite e compile o seguinte algoritmo:

1.1- Explique por palavras suas o que entendeu do algoritmo.

Algoritmo que soma 5 aos números em posição impar do vector e multiplica por 5 os números em posição par no vector.

```
#include<stdio.h>
#include<conio.h>
main()
{
int veca[10], vecb[10];
int i;
clrscr();
for (i=1;i<=5;i++)
    {
        printf("digite um valor para o %dª ",i);
        scanf("%d",&veca[i]);
    }
for (i=1;i<=5;i++)
    {
        if(i%2==0)
            vecb[i]=veca[i]*5;
        else
            vecb[i]=veca[i]+5;
    }
for (i=1;i<=5;i++)
    printf("veca[%d]=%d  vecb[%d]=%d\n",i,veca[i],i,vecb[i]);
scanf("%d");
}
```

- Neste algoritmo declaramos 2 vectores com 10 posições cada um, veca e vecb.

1	2	3	4	5	6	7	8	9	10
i	I+1	I+2	I+3	I+4	I+5	I+6	I+7	I+8	I+9

- No primeiro for vamos pedir ao utilizador que digite números para preencher 5 "células" do nosso vector. Supondo que o utilizador digita os números em baixo indicados o nosso vector ficará:

1	2	3	4	5	6	7	8	9	10
i	I+1	I+2	I+3	I+4	I+5	I+6	I+7	I+8	I+9
3	7	12	16	21	--	--	--	--	--

- No segundo for vamos multiplicar por 5 todos os números que ficarem numa posição par do nosso vector e vamos somar 5 a todos aqueles que ficarem numa posição impar. Assim:

1	2	3	4	5	6	7	8	9	10
i	I+1	I+2	I+3	I+4	I+5	I+6	I+7	I+8	I+9
3	7	12	16	21	--	--	--	--	--
3+5=8	7*5=35	12+5=17	16*5=80	21+5=26					

- No terceiro e último for fazemos o output do resultado.

Exercício 2

Altere o algoritmo anteriormente criado para que faça exactamente o mesmo mas desta vez deve utilizar como referencia os números do vector e não a posição dos números no vector.

```
#include<stdio.h>
#include<conio.h>
main()
{
int veca[10], vecb[10];
int i;
clrscr();
for (i=1;i<=5;i++)
    {
    printf("digite um valor para o %dª ",i);
    scanf("%d",&veca[i]);
    }
for (i=1;i<=5;i++)
    {
    if(veca[i]%2==0)
    vecb[i]=veca[i]*5;
    else
    vecb[i]=veca[i]+5;
    }
for (i=1;i<=5;i++)
    printf("veca[%d]=%d  vecb[%d]=%d\n",i,veca[i],i,vecb[i]);
scanf("%d");
}
```

Exercício 3

Utilizando vectores elabore um algoritmo que peça ao utilizador para introduzir 5 números e some todos os números em **posição** impar no vector.

```
#include<stdio.h>
main()
{
int a[5];
int soma,i;
soma=0;
for (i=1;i<=5;i++)
    {
    printf("digite o %dª valor",i);
    scanf("%d",&a[i]);
    }
for (i=1;i<=5;i++)
    {
    if (i%2!=0)
    soma=soma+a[i];
    }
printf("a soma , -> %d",soma);
scanf("%d");
}
```

Exercício 4

Utilizando vectores elabore um algoritmo que peça ao utilizador para introduzir 5 números e some todos os números impares do vector.

```
#include<stdio.h>
main()
{
int a[5];
int soma,i;
soma=0;
for (i=1;i<=5;i++)
{
printf("digite o %d$ valor",i);
scanf("%d",&a[i]);
}
for (i=1;i<=5;i++)
{
if (a[i]%2!=0)
soma=soma+a[i];
}
printf("a soma , -> %d",soma);
scanf("%d");
}
```

Exercício 5

Utilizando vectores elabore um algoritmo que peça ao utilizador para introduzir 5 números e apresente o resultado da soma dos números impares e pares do vector.

```
#include<stdio.h>
main()
{
int a[10];
int soma,soma2,i;
soma2=0;
soma=0;
for (i=1;i<=5;i++)
{
printf("digite o %d$ valor",i);
scanf("%d",&a[i]);
}
for (i=1;i<=5;i++)
{
if (a[i]%2!=0)
soma=soma+a[i];
else
soma2=soma2+a[i];
}
printf("a soma dos impares , -> %d",soma);
printf("\na soma dos pares , -> %d",soma2);

scanf("%d");
}
```

6- Digite e compile o seguinte algoritmo:

6.1- Explique por palavras suas o que entendeu do algoritmo.

Exemplo com vectores:

```
#include <stdio.h>
main ()
{
int num[100]; /* Declara um vector de inteiros de 100 posições */
int count=0;
int totalnums;
do
    {
    printf ("\nDigite o numero (-999 p/ terminar): ");
    scanf ("%d",&num[count]);
    count++;
    } while (num[count-1]!=-999);
totalnums=count-1;
printf ("\n\n\n\t Os números que digitou foram:\n\n");
for (count=0;count<totalnums;count++)
    {
    printf (" %d",num[count]);
    }
}
```

No exemplo acima, o inteiro *count* é inicializado a 0. O programa pede pela entrada de números até que o utilizador digite a **Flag** -999. Os números são armazenados no vector **num**. A cada número armazenado, o contador do vector é incrementado para na próxima iteração escrever na próxima posição do vector. Quando o utilizador digita a flag, o programa abandona o primeiro loop e armazena o total de números gravados. Por fim, todos os números são impressos. É bom lembrar que nenhuma restrição é feita quanto a quantidade de números digitados. Se o utilizador digitar mais de 100 números, o programa tentará ler normalmente, mas o programa vai escreve-los numa parte não criada de memória, pois o espaço criado foi somente para 100 inteiros. Isto pode resultar nos mais variados erros, principalmente em tempo de execução.