Programação Imperativa – Folha das aulas práticas nº 6

Temas: Exercícios variados com vectores.

Nome____ **Ncd**____________

1. *Ai, as notas...*
   Um vector `v[]` contém todas as notas obtidas até um determinado momento por um aluno. Pretende-se um programa que imprima as 2 notas mais elevadas (que podem ser iguais) já obtidas pelo aluno.
   Nota. O seu programa de teste pode ter a seguinte estrutura:

```
main(){
  int v[] = {8,8,10,7,18}, ...
  double r;
  ...
  printf("Notas mais altas = %2d, %2d\n",a,b);
}

Resultado esperado:
     Notas mais altas = 18 10
```

2. *Ontem e amanhã...*
   As temperaturas em `m` dias sucessivos (a uma determinada hora) estão registadas nos elementos de um vector `v`. Escreva uma função

   `void antes_e_depois(int m, int v[])`

   que tem por efeito substituir cada `v[i]` pela média das temperaturas do dia anterior e do dia seguinte. O primeiro e o último valor não são alterados. A única variável indexada que pode utilizar é `v`.
   Nota. O seu programa de teste pode ter a seguinte estrutura:

```
void antes_e_depois(int m, double v[]){
  ...
}

//-- Mostra os n primeiros elementos (floating point) de a[]
void mostra(int n,double a[]){
  int i;
  for(i=0;i<n;i++)
    printf("%8.1f",a[i]);
  printf("\n");
}

main(){
  double temp[]={10.0, 10.2, 11.8, 31.4, 2.8}, n=5;
  mostra(n,temp);
  antes_e_depois(n,temp);
  mostra(n,temp);
}
```

   Para este teste o resultado esperado é

```
    10.0    10.2    11.8    31.4     2.8
    10.0    10.9    20.8     7.3     2.8
```

   Note-se, por exemplo, que `21.6` é a média entre `11.8` e `31.4`.

3. *De texto para número...*
Pretende-se uma função

`int conv(char *s,int b)`

que converta a primeira sequência de dígitos do "string" `*s` (dígitos de um inteiro escrito na base `b`, $\mathtt{b} \leq 10$) para um inteiro e o retorne; se não existir nenhum dígito em `s`, a função deverá retornar -1. Por exemplo, se

$$\mathtt{s} = \text{"}\mathtt{Ai:}\overline{1011}!1!\text{"}, \mathtt{b} = 2$$

deve ser retornado $11 = 1011_2$.
Nota. O seu programa de teste pode ter a seguinte estrutura:

```
#define MAX 100

int conv(char *s,int b){
  ...
}

int main(){
  char s[MAX];
  int r;
  do{
    printf("? (0 para acabar)  "); scanf("%s",s);
    r=conv(s,3);
    if(r<0)
      printf("Formato errado\n");
    else
      printf("Valor = %d\n",r);
  }
  while(r!=0);
}
```