

Programação Imperativa – Folha das aulas práticas nº 2 (exercícios extra)

**Temas:** CONVERSÕES DE BASE (INTRODUÇÃO), CÁLCULO DE MÉDIAS, DIVISIBILIDADE (NÚMEROS PERFEITOS).

Nome \_\_\_\_\_ Ncd \_\_\_\_\_

1. *Muda-me a base!*

Com este exercício pretende-se mudar para a base 10 um inteiro dado na base 2, dígito a dígito. O comportamento do programa deve ser do seguinte tipo

```
Num. de bits? 4
1
1
0
0
Numero na base 10: 12
```

Note-se que  $1100_2 = 12$ .

**Sugestão.** Note que o número do exemplo pode ser calculado de forma iterativa, multiplicando sempre o valor actual por 2 e somando-lhe 0 ou 1, conforme o bit lido.

$$\begin{array}{c} \overbrace{\hspace{10em}}^{12} \\ \overbrace{\hspace{6em}}^6 \\ \overbrace{\hspace{3em}}^3 \\ \overbrace{\hspace{1em}}^1 \\ (((\boxed{1} \times 2) + \boxed{1}) \times 2 + \boxed{0}) \times 2 + \boxed{0} = 12 \end{array}$$

```
//-- Muda um inteiro dado da base 2 para a base 10
main(){
    int n, //-- inteiro a obter (base 10)
        m, //-- numero de bits do inteiro dado
        b, //-- inteiros lidos (0 ou 1)
        i; //-- contador (para se ler b m vezes)
    n=0;
    printf("Num. de bits? "); scanf("%d",&m);
    for(i=0;i<m;i++){
        scanf("%d",&b);
        n=n*2+b;
    }
    printf("Numero na base 10: %d\n",n);
}
```

## 2. Médias...

- (a) Escreva um programa com o seguinte objectivo:

**Dados:**  $n$  (inteiro não negativo) seguido de  $n$  inteiros.

**Resultado:** A média dos  $n$  elementos lidos, uma vez desprezados o maior e o menor desses elementos; supõe-se que esses elementos são únicos e distintos e que a média está definida.

**Exemplo:** Para os dados 4,2,6,8,5 (4 é o número de elementos a ler) é impresso 5.5.

**Sugestão:** À medida que os elementos são lidos, guarde a soma deles, o maior já encontrado e o menor já encontrado por exemplo nas variáveis  $s$ ,  $max$  e  $min$ . As 2 últimas instruções do seu programa podem ser

```
med=((double)(s-min-max))/(n-2);
printf("Media = %lf\n",med);

main(){
  int s, n, i, x, min, max;
  double med;
  scanf("%d",&n);
  scanf("%d",&x);
  max=x;
  min=x;
  s=x;
  for(i=2;i<=n;i++){
    scanf("%d",&x);
    s+=x;
    if(x>max) max=x;
    else
      if(x<min) min=x;
  }
  med=((double)(s-min-max))/(n-2);
  printf("Media = %lf\n",med);
}
```

- (b) Semelhante à alínea anterior mas não se supondo que o maior e o menor dos elementos são únicos.

**Exemplo:** Para os dados 7,2,8,6,2,8,5,5 é impresso 5.333.

**Sugestão:** Guarde *também* o número de vezes que o maior e o menor dos elementos já lidos ocorreu. Por exemplo, para os dados seguintes e na altura indicada por "\*", esses valores são respectivamente 3 e 1 uma vez que o máximo até essa altura (5) ocorreu 3 vezes e o mínimo até à altura (1) ocorreu 1 vez

```
8 | 2  5  5  1  5  9  1  3
      *
      *
```

Após a leitura de todos os elementos, esses valores são 1 e 2, respectivamente (porquê?).

PORQUE O MÁXIMO (9) OCORRE 1 VEZ E O MÍNIMO (1) 2 VEZES.

```
main(){
  int s, n, i, x, min, max, cmin,cmax;
  double med;
  scanf("%d",&n);
  scanf("%d",&x);
  max=x; cmax=1; min=x; cmin=1; s=x;
  for(i=2;i<=n;i++){
    scanf("%d",&x);
    s+=x;
    if(x>max) {max=x;cmax=1;}
    else if(x<min) {min=x;cmin=1;}
    else{if(x==max) cmax++;if(x==min) cmin++;}
  }
  med=((double)(s-min*cmin-max*cmax))/(n-cmin-cmax);
  printf("Media = %lf\n",med);
}
```

### 3. Perfeito?

(a) Escreva uma função

```
int perfeito(int n)
```

que retorna 1 se o inteiro  $n$  é igual à soma dos seus divisores com exclusão do próprio número (exemplo:  $6=1+2+3$ ) e retorna 0 caso contrário.

```
int perfeito(int n){
    int s=0,i;
    for(i=1;i<=n/2;i++){ // para cada candidato a divisor i
        if(n%i==0) s+=i; // se i divide n soma i a n
        if(s>n) return(0); // soma parcial > n => n não é perfeito
    }
    return(s==n); // retorna o valor "booleano"
}
```

---

(b) Escreva um programa que use a função da alínea anterior para imprimir os inteiros perfeitos não superiores a 10 000.

```
#define MAX 10000

int perfeito(int n) // ver a resposta à alínea anterior...

int main(){
    int i;
    for(i=1;i<=MAX;i++) // para cada i, 1 <= i <= MAX
        if(perfeito(i)) // se i é perfeito...
            printf("%d\n",i); // escreve i
}
```

---