Programação Imperativa – Folha das aulas práticas nº 4

Temas: Exercícios variados - Uma Resolução

Nome____**Ncd**____________

1. Operações com conjuntos...
Um conjunto é representado pela sua cardinalidade e por um vector (não necessariamente ordenado) contendo os seus elementos, sem repetições. Por exemplo, o conjunto $\{2, 6, 8, 9\}$ *pode* ser representado por

   `m=4,  v[] = [9,6,2,8]`

   (isto é, o conjunto tem 4 elementos, `v[0]=9`, etc.).

   (a) Escreva uma função

   `int correcto(int m, int v[])`

   que retorna 1 se os elementos do vector `v[]` são distintos e 0 caso contrário (use uma pesquisa sequêncial).

```
int correcto(int m, int v[]) {
  int i,j;

  for(i=0;i<m-1;i++)
    {
      for(j=i+1;j<m;j++)
        {
          if(v[i]==v[j])
            return 0;
        }
    }
  return 1;
}
```

   (b) Usando a função `pertence`, escreva uma função que retorna o número de elementos pertencentes à reunião dos conjuntos representados por (`m`, `v`) e por (`k`, `w`)

   `int reunião(int m, int v[],int k,int w[])`

   Por exemplo para

```
m=3,  v=[8,3,5]
k=6,  w=[10,5,9,3,1,4]
```

   o resultado retornado deve ser `6` (porquê?)
   Nota. Pode experimentar as funções que escreveu com o seguinte programa principal que inclui já a função pertence

```
//-- n pertence ao conjunto representado por (m,v)?
int pertence (int n, int m, int v[])
{
  int i=0;
  while (i<m && v[i]!=n)
    i++;
  return (v[i]==n);
}

int reuniao(int m,int v[],int k,int w[]){
  int i,iguais=0;

  for(i=0;i<m;i++)
    iguais+=pertence(v[i],k,w);
  return (k+m-iguais);

}
```

```
main(){
  int v[]={8,9,3}, w[]={10,5,9,1,3};
  printf("%d elementos na reuniao\n",reuniao(3,v,5,w));
}
```

2. Corta o inicio...
   Escreva um programa que elimine todas as letras iniciais das palavras, considerando-se que uma palavra é uma sequência máxima de letras.

   `....+ Ah, que Ficheiro!!!` ⟶ `....+ h, ue icheiro!!!`

   A leitura de dados e a escrita de valores deve ser efectuada exclusivamente com `getchar` e `putchar`.
   Posteriormente estenda o seu programa para letras acentuadas e com cedilhas (começe por determinar experimentalmente – com um programa – os códigos dessas letras).

```
#include<stdio.h>

int main(){
  char c,dentro=1;

  c = getchar();
  while (c!=EOF) {
    if ( c == ' ' || c == '\n' ) dentro = 0;
    c = getchar();
    if (dentro) printf("%c",c);
    dentro = 1;
  }

  return 0;
}
```

---

3. Estranho...
   O que imprime o seguinte programa?

```
char *s="char *s=%c%s%c%;%cmain(){printf(s,34,s,34,10,10);}%c";
main(){printf(s,34,s,34,10,10);}
```

   Nota. Os inteiros 34 e 10 são os códigos ascii de respectivamente "`"`" (aspas) e de "nova linha"("line feed"). Vai descobrir algo de muito curioso[1]. Depois, certifique-se da sua resposta, experimentando no computador!.

```
char *s="char *s=%c%s%c;%cmain(){printf(s,34,s,34,10,10);}%c";
main(){printf(s,34,s,34,10,10);}
```

[1]... e relacionado com o teorema de incompletude de Gödel.