Instituto Superior de Engenharia de Coimbra
Departamento de Engenharia Informática e Sistemas

Introdução à Programação

Ficha Laboratorial 6

Tópicos da matéria:

Utilização de *arrays* em linguagem C.

Bibliografia:

"Linguagem C": Luis Damas
Capítulo 6.

"C programming: A Modern Approach": K. N. King
Capítulo 8.

***Nota*: *Antes da implementação deve desenvolver o algoritmo* para cada um dos exercícios propostos.**

*Array*s unidimensionais

1. Desenvolva um programa que calcule a média dos N elementos de um array de inteiros. A inicialização dos elementos do array deverá ser feita pelo utilizador.

2. Complete o programa anterior para que todos os elementos do array, cujo valor seja inferior à média, sejam colocados a zero.

3. Um array pode servir para representar conjuntos de números inteiros. Considerando o domínio [0, 9], um array de 10 elementos pode indicar quando é que cada um dos números pertence ou não a um determinado conjunto. Para isso, basta colocar o valor de a[i] a 0 se o elemento i não fizer parte do conjunto e a[i] a 1, se i fizer parte do conjunto.
O array

1	0	1	1	0	1	1	1	0	0

indica que os elementos 0, 2, 3, 5, 6, 7 fazem parte do conjunto.

Desenvolva um programa que, após obter e armazenar dois conjuntos A, B com domínio [0, 9], calcule as seguintes operações: A $\cap$ B (intersecção de A com B), A $\cup$ B (reunião de A com B) e A – B (elementos de A que não pertencem a B).

4. Desenvolva uma função, que receba dois arrays de inteiros (e respectiva dimensão) e verifique se estes são iguais. A função deve devolver 1 se os dois arrays forem iguais ou devolver 0 se forem diferentes. Considere que dois arrays são iguais se, na mesma posição, tiverem elementos com o mesmo valor.

5. Desenvolva um programa que verifique se um determinado número inteiro positivo é capicua. O número a verificar é especificado pelo utilizador.

Nota: um número é capicua se for "simétrico", como por exemplo os números **12321** e **694496**.

6. Defina uma função que receba como argumentos um array de inteiros e a sua dimensão e devolva a soma de todas as ocorrências do maior número do array.

Exemplo:

Array: 1 2 8 8 1 8	**devolve : 24**
Array: 2 2 2 2 2 2	**devolve : 12**
Array: 1 2 3 2 -1 0	**devolve : 3**

Usando a função faça um programa que:

- Declare um array de inteiros de tamanho TAM.
- Peça ao utilizador valores inteiros para preencher o array.
- Chame a função anterior enviando como argumentos o array e a sua dimensão.
- Imprima o valor devolvido pela função.

7. Considere um array com N componentes inteiros.

a) Desenvolva uma função que devolva a posição do maior dos elementos do array. No caso de haver repetições do valor no array, a função pode, por exemplo, fornecer como resultado a posição com maior índice.

b) Desenvolva uma função que desloque todos os seus elementos, uma posição para a direita. O último elemento deve deslocar-se para a primeira posição.

c) Elabore um programa que, após a leitura das N componentes inteiras, faça as necessárias *rotações* para a direita até que o elemento de maior valor do array se encontre na última posição desse array.

8. Os elementos de um array dizem-se ordenados em pirâmide se o seu valor aumentar até uma certa posição e a partir daí diminuir. Dois possíveis exemplos são:

{1, 2, 4, 5, 6, 3, 2, -23, -23, -120} ou {1, 20, 19, 18, 17, 16, 15, -100, -101, -102}

a) Declare um array local à função main() com capacidade para armazenar DIM_TAB números inteiros (sendo DIM_TAB uma constante simbólica);

b) Considere que o *array* declarado na alínea anterior já foi completamente preenchido com números inteiros ordenados em pirâmide. Desenvolva uma função que receba o *array* (e respectiva dimensão) por argumento e escreva no monitor os valores dos seus elementos, por ordem decrescente.

9. Escreva uma função que receba como argumentos dois arrays VC e VI de dimensão n, sendo o primeiro um array de caracteres e o segundo um array de inteiros. A função deve escrever no ecrã cada elemento de VC (um caracter), repetindo-o um nº de vezes igual ao valor inteiro guardado no elemento correspondente de VI. Cada elemento de VC deve ser escrito numa nova linha. Protótipo da função:

void func(char VC[], int VI[], int n);

10. Numa determinada escola irá decorrer um referendo sobre a proibição, ou não, de fumar no recinto da mesma. Com o objectivo de realizar uma sondagem sobre o resultado da votação, um grupo de docentes resolveu efectuar um pequeno inquérito anónimo a N alunos, escolhidos criteriosamente. Assim cada aluno tinha unicamente que indicar, com uma cruz, uma das seguintes opções, como resposta à pergunta "É a favor da proibição de fumar no recinto da escola?":

SIM, NÃO, ESTOU INDECISO, VOU ABSTER-ME.

Desenvolva um programa que auxilie os docentes na obtenção dos resultados da sondagem. Assim, para cada um dos N inquéritos, o programa deve pedir ao utilizador um carácter indicativo da resposta do aluno, sendo 'S' para a resposta SIM, 'N' para NÃO, 'I' para INDECISO e 'A' para VOU-ME ABSTER.

No caso do aluno não ter assinalado nenhuma opção, ou mais do que uma, o carácter a introduzir pelo utilizador deverá ser 'X'. No final da introdução dos dados, o programa deve indicar qual será, previsivelmente, o resultado do referendo e, para além disso, disponibilizar os totais de cada uma das opções sob um formato gráfico, recorrendo à função definida na alínea anterior.

Exemplo de execução:

```
 Indique total de inquéritos: 50
--->1º Inquérito
 -----> Opção escolhida (S/N/I/A/X): N
--->2º Inquérito
-----> Opção escolhida (S/N/I/A/X): S
--->3º Inquérito
-----> Opção escolhida (S/N/I/A/X): S

(...)

--->50º Inquérito
 -----> Opção escolhida (S/N/I/A/X): I

Segundo a sondagem a opção vendedora será a SIM.
Resultados das várias opções:

SSSSSSSSSSSSSSSSSS
NNNNNNNNN
IIIIIIIIIII
AAAAAA
XXX
```

Nota: O programa deverá fazer a validação dos dados introduzidos pelo utilizador.

11. No Banco BAP cada um dos clientes tem um número de identificação único, denominado NIP (número de identificação pessoal). O NIP de cada cliente tem a seguinte informação:

- **país de origem**: código alfanumérico com PAIS caracteres que indica a que país pertence o cliente;
- **número de conta**: sequência de NUM_CONTA dígitos que armazena o número da conta do cliente;
- **tipo de conta**: sequência de TIPO_CONTA dígitos que indica o tipo de conta do cliente;
- **espaço livre**: sequência de ESP_LIVRE dígitos não utilizados;

Quando o sistema informático necessita de aceder ao NIP de um determinado cliente armazena-o num array de caracteres local à função main() com a seguinte declaração:

```
void main(void)
{
    char nip_actual[PAIS+NUM_CONTA+TIPO_CONTA+ESP_LIVRE];
    ...
}
```

Exemplo do NIP de um cliente do país **PT1**, número de conta **235812** e tipo **300**:

P	T	1	2	3	5	8	1	2	3	0	0			
País de origem			Número de conta						Tipo de conta			Espaço não utilizado		

Para responder às questões que se seguem, considere que o NIP do cliente já está armazenado na estrutura de dados nip_actual, e que as constantes simbólicas PAIS, NUM_CONTA, TIPO_CONTA e ESP_LIVRE já se encontram definidas.

a) Verificou-se que PAIS caracteres para o código alfanumérico que identifica o país do cliente são insuficientes. É necessário adicionar dois novos caracteres ao final desta sequência (adicionando igualmente o valor 2 à constante PAIS e subtraindo 2 à constante ESP_LIVRE). Para actualizar a informação do NIP, toda a informação que se segue ao campo *país* deve ser deslocada duas posições para a direita. Desenvolva uma função que efectue essa operação no NIP que recebe por parâmetro. A função recebe como argumentos o NIP a alterar e os novos caracteres a adicionar (os caracteres são passados como dois argumentos separados). Assuma que as constantes ainda não foram actualizadas.

   Considerando o exemplo anterior, se a função receber o caracteres 'Q' e '2', o novo NIP ficará:

P	T	1	Q	2	2	3	5	8	1	2	3	0	0	

b) O banco tem a informação de que um dos seus clientes é um perigoso falsário. A única informação de que dispõe para o identificar é que o código do seu país de origem contém o caracter 'R' e que a sequência de dígitos '456' aparece no número de conta. Desenvolva uma função que verifique se o NIP passado por argumento obedece a estas duas condições. A função deve devolver 1 se isso suceder, e 0 no caso contrário.

12. Para processar os dados de cada apostador do totoloto, a Santa Casa da Misericórdia armazena temporariamente o seu nome e os números da sua aposta num array de caracteres local à função main com a seguinte definição. (Considere que a constante simbólica TAM_INFO já se encontra definida)

```
char info[TAM_INFO];
```

Quando estão armazenadas no array, as várias palavras que constituem o nome e os números da aposta estão separadas entre si por um ou mais espaços em branco. Um exemplo possível da informação armazenada é o seguinte:

“ Jorge Manuel Campos Sousa 23 30 32 34 39 44 ”

Para responder à questão seguinte, considere que a informação respeitante a um apostador já está armazenada no *array* info. A informação é terminada com um ‘\0’.

Desenvolva uma função que receba por parâmetro o *array* info e escreva no monitor os seis números em que o apostador armazenado no *array* info apostou. A função deve escrever um número por linha. A ordem pela qual os números são escritos é irrelevante.

Arrays multi-dimensionais

13. Desenvolva um programa que inicialize um *array* 10×10 com valores aleatórios entre 0 e 10 (para isso, pode utilizar a função apresentada no exemplo 13 da Folha 5). Após a inicialização, o programa deve contar o número de zeros existentes no *array*.

14. Desenvolva um programa que inicialize um *array* 10×3 da seguinte forma: em cada uma das linhas, a primeira coluna deve ficar com um inteiro entre 1 e 100 introduzido pelo utilizador, a segunda coluna com o quadrado deste valor e a terceira com o cubo. Após a inicialização, o programa deve contar quantas posições do *array* têm valores superiores a 1000.

15. Um treinador de atletismo treina 3 atletas e faz 5 sessões de treino por semana. Em cada sessão, cada atleta percorre uma distância que é cronometrada. Os valores dos tempos, em segundos, são registados sob a forma de uma matriz T(3, 5), onde cada linha diz respeito a um atleta e cada coluna a uma sessão de treino.

Escreva um programa para:

a) Calcular e escrever a média dos tempos realizados em cada sessão de treinos.

b) Determinar o melhor tempo realizado por cada um dos atletas nas 5 sessões.

16. Na matriz IP(24, 6) de elementos inteiros, encontram-se registadas as notas dos 24 alunos de cada uma das 6 turmas de Introdução à Programacão. Sabendo que todos os elementos de IP são valores entre 0 e 20, elabore um programa que determine e escreva:

a) o número de alunos aprovados (nota ≥ 10).

b) a melhor nota em cada uma das turmas.

c) a turma com maior número de alunos aprovados.

17. Um turista pretende efectuar uma viagem seguindo os seguintes percursos:

Origem	Destino
A	**B**
B	**C**
C	**D**
D	**E**
E	**F**
F	**A**

Para tal, consultou seis agências de viagens e, em cada uma delas, soube o preço de cada um dos percursos. Com estas informações, construiu uma matriz P(6,6).

a) Elabore um programa que determine qual a agência que oferece o preço mais baixo para a viagem de acordo com a ordem dos percursos que o turista pretende efectuar.

b) Considere agora, que cada uma das agências promove, a um preço mais baixo, do que o de qualquer outra das restantes cinco agências, um dos seis trajectos acima descritos. Com base na matriz P(6,6) e na ordem pretendida para os percursos, elabore um programa que construa uma matriz V(6,6) em que a diagonal de V representa a viagem que o turista pretende efectuar ao preço mais baixo.

18. Considere a seguinte tabela:

	A	B	C	D
A	0	-1	12	1
B	-1	0	5	6
C	12	5	0	-1
D	1	6	-1	0

A informação da tabela representa as distâncias (utilizando uma ligação directa) entre 4 cidades (A, B, C, D). Quando aparece a informação -1, as cidades não estão ligadas directamente.

a) Declare uma estrutura de dados local à função main que permita armazenar a informação da tabela.

b) Desenvolva uma função que receba a tabela e o nome de uma das cidades e devolva a distância a que a cidade mais próxima se encontra (ligação directa).