

Programação Imperativa 2022/2023 (CC1003), DCC/FCUP

Folha 5

5.1 (Plataforma codex) Defina a função

```
void range(int vec[], unsigned size, int inicio, int incr)
```

para inicializar os primeiros `size` elementos de um vector (array) `vec` com os valores `inicio`, `inicio+incr`, `inicio+2*incr`, etc., isto é, segundo uma progressão aritmética, com termo inicial `inicio` e razão `incr`. Exemplo:

```
int a[5];  
range(a, 5, 3, 2); // a[] passa a conter {3, 5, 7, 9, 11}
```

5.2 Escreva uma função `void capitalizar(char str[])` que transforma todas as letras de uma cadeia em maiúsculas; os outros caracteres devem ficar inalterados. O argumento é uma cadeia de caracteres (não necessariamente letras) terminada por `'\0'` (o valor 0).

5.3 Escreva uma função `int palindromo(char str[])` que testa se uma cadeia de caracteres é um palíndromo, isto é, se é igual quando lida da esquerda para a direita ou da direita para a esquerda.

5.4 Escreva uma função `int todos_letras(char str[])` que testa se uma cadeia contém apenas letras (maiúsculas ou minúsculas). O resultado deve ser 1 em caso afirmativo e 0 em caso negativo. Pode resolver este exercício recorrendo a funções da biblioteca `ctype.h`.

5.5 Escreva uma função `int algum_digito(char str[])` que testa se uma cadeia contém algum dígito decimal (isto é, algum dos valores `'0'`, `'1'`, ..., `'9'`). O resultado deve ser 1 em caso afirmativo e 0 em caso negativo.

5.6 (Plataforma codex) Escreva uma função `int forte(char str[])` que verifica se uma cadeia de caracteres é uma “*password* (palavra-passe) forte” usando o seguinte critério:

- Deve ter pelo menos 6 caracteres;
- Deve conter pelo menos uma letra maiúscula, uma letra minúscula e um algarismo.

O resultado da função deve ser 1 se ambos os critérios se verificam e 0 caso contrário. Por exemplo: “Abr4cadabra” e “Apric0t” são palavras-passe fortes, mas “Ub40” não é (porque o comprimento é inferior a 6 caracteres) e “POLICE” também não (porque só tem letras maiúsculas).

5.7 Escreva uma função `int decimal(char str[])` que dada uma cadeia de caracteres que contém apenas dígitos (`'0'`, `'1'`, ..., `'9'`) além do terminador `'\0'`, a qual representa um inteiro não negativo em decimal (i.e., na base 10), retorna o valor desse inteiro. Por exemplo: `decimal("1234")` deve retornar o inteiro 1234.

5.8 Escreva uma função `int contar_maiores(int vec[], int size, int val)` cujos argumentos são uma variável indexada `vec` com tamanho `size` e um valor `val` e cujo valor de retorno deve ser a contagem do número de elementos de `vec` que são estritamente maiores do que `val`.

5.9 (Plataforma codex) Escreva uma função `int filtrar_positivos(int vec[], int size)` que remove os valores não positivos (isto é, negativos ou zero) de um vector `vec` com tamanho `size`. A função deve modificar a variável indexada dada de forma a que os valores positivos fiquem num segmento inicial do vector. O valor de retorno deve ser o número de valores positivos (i.e., o comprimento do segmento inicial).

5.10 (Plataforma codex) Escreva uma função `int repetidos(int vec[], int size)` que testa se há (pelo menos) dois valores iguais no vector `vec` com tamanho `size`; o valor de retorno deve ser 1 em caso afirmativo e 0 em caso negativo. Exemplos:

```
int a[5] = { 2, -1, 0, 2, -1 };
int b[5] = { 3, 4, 1, 2, -1 };
printf("%d\n", repetidos(a, 5)); // imprime 1
printf("%d\n", repetidos(b, 5)); // imprime 0
```

A sua função não deve modificar os elementos do vector (array) passado como argumento.

5.11 Numa turma com 25 alunos, qual é a probabilidade de que o aniversário de pelo menos dois deles seja no mesmo dia? E se forem 50 alunos? O *paradoxo do aniversário* é a constatação de que esta probabilidade é maior do que parece ao senso comum¹. Vamos estimar experimentalmente a probabilidade de aniversários repetidos com N alunos escrevendo um programa que repetidamente gere dias aleatoriamente (N valores de 1 a 365) e verifique a ocorrência de repetições. Utilize `rand()` para gerar aleatoriamente um vector com N dias e a função do exercício 5.10 para testar repetições. Se fizer um grande número de experiências, a frequência relativa de ocorrência de repetidos dá um valor aproximado da probabilidade. O número de alunos N e de experiências deve ser facilmente modificável (por exemplo, usando `#define`).

¹https://en.wikipedia.org/wiki/Birthday_problem