

MEFT - Programação

1º Ano - 1º Semestre de 2018/2019

Série 2 (22/10/2018)

1. Reescreva o problema 2 da série 1, de tal maneira que:

- a) Se possam dar os valores de '**N**' e de '**Max**' na linha de comandos;
- b) Se o utilizador não der argumentos, então o programa deve pedi-los;
- c) A atribuição de memória ao vector seja feita usando a função '**malloc**' e usando apenas o tamanho exigido;
- d) No final, deverá perguntar se quer voltar a executá-lo, no caso afirmativo, pedir ao utilizador quais os novos valores '**N**' e '**Max**' e fazer o ajuste da atribuição de memória, para o novo valor mínimo, usando a função '**realloc**';
- e) Ao encerrar o programa deve ser libertada a memória (função '**free**').

2. A função *sin* pode ser obtida a partir do seu desenvolvimento em série de Taylor:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2*i+1}}{(2*i+1)!}$$

Sendo uma série alternada decrescente, em valor absoluto, que converge para zero, cada termo que se adiciona à série, pode ser considerado como um majorante do erro para essa ordem.

Pretende-se construir um programa que pede ao utilizador o valor do argumento '*x*', que calcula o valor de '*sin(x)*' com um erro inferior a 10^{-10} e que mostra o resultado obtido com a função que escreveu e o resultado da função '*sin*' da biblioteca de **C** com pelo menos 10 casas decimais.

Para tal, o programa deverá conter as seguintes funções:

- a) Uma função que converte o argumento '*x*' para o intervalo $[0, 2\pi[$. Poderá usar a função '**fmod**' e, no caso de ser negativo, fazer em seguida o devido ajuste;
- b) Uma função que calcule o factorial de um número (utilize '**double**' para o resultado do cálculo);
- c) Uma função que recebe o valor do argumento '*x*' e que retorna o valor do '*sin(x)*', com um erro inferior ao referido (10^{-10}), calculado a partir do desenvolvimento em série de Taylor acima descrito.

Nota 1: A função **fmod** retorna, em '**double**'s, o resto da divisão (ver manual de C):

double fmod (double numerator, double denominator).

Para ver o seu funcionamento da função experiente-a com valores positivos e negativos.

Nota 2: O valor de π encontra-se definido em '**math.h**' e é dado por '**M_PI**'.

(v.s.f.f.)

3. Construa um programa que lê dois argumentos da linha de comando. A partir dessas leituras, deverá executar as seguintes tarefas:

- a) Imprima no ecrã as duas 'strings' contidas naqueles dois argumentos e diga quais são os seus comprimentos;
- b) Com a função 'malloc' crie uma nova 'string' para a qual irá copiar o primeiro argumento e a seguir acrescentar o segundo. Deverá alocar apenas o número de bytes necessários para que a tarefa seja correctamente efectuada. No final mostre o resultado no ecrã;
- c) Repita a alínea 'b)' mas copiando primeiro o segundo argumento e depois acrescentando o primeiro;
- d) Compare as duas strings criadas nas alíneas 'b)' e 'c)' usando a função 'strcmp' e apresente o resultado obtido dizendo qual a maior segundo o critério daquela função.

4. Pretende-se construir um programa que recebe duas informações horárias (h:m:s), na linha de comandos e guarda cada uma delas numa estrutura. Pretende-se:

- a) Verificar, usando as estruturas, se as informações horárias estão correctamente escritas (horas de 0 a 23 e minutos e segundos de 0 a 59) e, caso não estejam, o diga;
- b) Imprimir cada uma das estruturas que contêm as informações horárias;
- c) O número de segundos, desde o início do dia, de cada uma delas;
- d) O número de minutos, desde o início do dia, de cada uma delas, na representação de ponto flutuante;
- e) O resultado da sua soma ignorando a mudança de dia (o resultado deve ainda ser obtido usando as estruturas anteriores e escrito numa nova estrutura);

Nota: A estrutura a usar deve ter três membros inteiros (hora, minutos e segundos).