

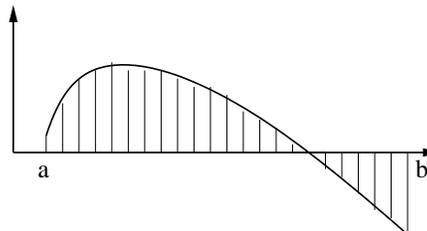
MEFT - Programação

1º Ano - 1º Semestre de 2019/2020

Série 7 (25/11/2019)

1. Integral definido de uma função.

Para efectuar o cálculo do integral definido de uma função, existem diversos métodos mais ou menos sofisticados. Um dos mais simples consiste no cálculo da soma das áreas dos rectângulos de base Δx e altura igual ao valor da função num ponto desse intervalo (método do rectângulo).



Tal corresponde a uma versão finita da definição do integral de Riemann. Para a altura de cada rectângulo pode usar uma das variantes deste método, a do ponto médio, que consiste em calcular o valor da função no ponto médio do intervalo.

a) Construa uma função que recebe como argumento um ponteiro para a função a integrar e os limites de integração e retorna o valor do respectivo integral.

b) Escreva um programa que usa aquela função para calcular os integrais das seguintes funções:

$$\sin(x), \quad 5 \cos(2x) e^{-0.2*x}, \quad \sqrt{x}, \quad 7x^5 - 3x^2, \quad e^x, \quad \frac{1}{\sqrt{\pi}} e^{-x^2}$$

O utilizador deverá poder escolher através dum menu a função que deseja bem como os seus limites de integração.

Nota: Tenha em conta que a inversão dos limites de integração é dada por:

$$\int_B^A f(x) dx = - \int_A^B f(x) dx$$

2. Construa um programa que gera N números complexos aleatórios com valores (parte real e imaginária) no intervalo $[-8, 16]$. N deve ser fornecido ao programa na linha de comandos e o programa deve executar as seguintes tarefas:

a) Ordenar os complexos por ordem decrescente das suas normas e escrevê-los no ecrã (incluindo a sua norma);

b) Perguntar ao utilizador se deseja trocar a ordem de algum par de complexos e, no caso afirmativo, executar a troca, mostrá-los novamente no ecrã e voltar a fazer a pergunta;

Nota: Os complexos devem ser internamente organizados num **vector de ponteiros** para estruturas alocado de acordo com o valor N . A estrutura associada ao complexo deve ter a parte real, a parte imaginária e a sua norma. A troca dos complexos deve ser feita através da trocas dos ponteiros e não dos seus conteúdos. A ordenação e a troca devem ser feitas em funções separadas.

3. Escreva um programa que calcula todos os números primos até um certo valor máximo **N**, fornecido pelo utilizador na linha de comandos. Os números primos obtidos deverão ser impressos no ecran e organizados em 6 colunas. Escreva ainda os valores obtidos num ficheiro em formato binário.

Para os obter, pode ser usado o “**Crivo de Eratóstenes**” que consiste em:

- Cria-se um vector de **N+1** elementos e põem-se os seus valores num dado valor, por exemplo, em “1” (faça uma alocação para esse efeito).
- Determina-se o maior número a ser testado, que é igual à raiz quadrada do valor pretendido, e arredondado para baixo, ‘**Teste**’;
- Começando em “2” marcam-se como eliminados-se todos os múltiplos desse número até **N**, por exemplo, com “0”,
- Passa-se ao número seguinte não eliminado e voltam a eliminar-se os seus múltiplos;
- Repete-se o passo anterior até chegar a ‘**Teste**’;
- Os números primos são os números não eliminados começando em “2”.

Uma vez escrito o ficheiro, pergunte ao utilizador se quer ver algum número primo e, no caso afirmativo, pergunte-lhe o número de ordem do primo que deseja (considerando ‘2’ o primeiro primo), posicione-se no ficheiro para o valor desejado, faça a sua leitura e mostre o resultado no ecran. Finalmente, pergunte se deseja ver mais algum número e, no caso afirmativo, volte a perguntar.

Nota: Para ler, escrever e se posicionar num ficheiro em binário, tem ao dispor as seguintes funções (ver manual da biblioteca de C):

- `size_t fread` (void *data, size_t size, size_t count, FILE *stream)
- `size_t fwrite` (void *data, size_t size, size_t count, FILE *stream)
- `int fseek` (FILE *stream, long int offset, int whence)

Em que “whence” é a posição a partir da qual o *offset* é calculado:

1. **SEEK_SET**: a partir do início do ficheiro;
2. **SEEK_CUR**: a partir da posição actual no ficheiro;
3. **SEEK_END**: a partir do fim do ficheiro;

4. Construa um programa que lê, usando a função “fgetc”, um ficheiro de texto, dado como argumento ao programa, e que executa as seguintes contagens imprimindo-as no ecran (ignore as cedilhas e vogais acentuados):

- a) Número total de caracteres do texto;
- b) Número de vezes que cada caracter aparece no texto (imprimindo apenas os encontrados);
- c) Número de vogais do texto;
- d) Número de consoantes do texto;

Nota: Função ‘getc’:

```
int fgetc (FILE *stream)
```

Esta função lê de ‘stream’ um caracter como um “unsigned char” e retorna o seu valor convertido para um “int”. Quando chega ao fim do ficheiro, ocorre um erro de leitura e retorna EOF.