Estrutura de Dados e Laboratório de Programação II
Lista de Exercícios
Revisão de Algoritmos com C++

Parte 1 – Entrada e Saída

1. **Instrução `cout`.**

   Desenvolver um programa para imprimir um menu de opções na tela para um sistema de cadastro de alunos.

```
  =============== CADASTRO DE ALUNOS ==================
  [1] Incluir aluno
  [2] Excluir aluno
  [3] Alterar aluno
  [4] Sair
  ======================================================
```

2. **Formatação de valores.**

   Desenvolver um programa para imprimir o valor 2.346728 com 1, 2, 3 e 5 casas decimais. Em C++, para limitar o número de casas decimais de um número de ponto flutuante, basta incluir na instrução `cout`, antes do número que se deseja imprimir, o seguinte conjunto de instruções:

   ```
   setiosflags(ios::fixed) << setprecision(n) <<
   ```

   onde `n` é o número de casas decimais desejado.

   Exemplo:

```cpp
#include <iostream>
#include <iomanip> // necessário para usar setprecision e
    ↪ setiosflags

using namespace std;

int main()
{
    double f = 3.14159;
    cout << setiosflags(ios::fixed) << setprecision(2) << f <<
        ↪ endl;
    return 0;
}
```

3. **Instrução `cin`.**

   Desenvolver um programa para efetuar a leitura do número de quilowatts consumido e calcular o valor a ser pago de energia elétrica, sabendo-se que o valor a pagar por quilowatt é de 0,12. Apresentar o valor total a ser pago pelo usuário acrescido de 18% de ICMS.

Parte 2 – Estruturas de Controle

1. **Loops e condições.**

   A prefeitura de uma cidade fez uma pesquisa entre seus habitantes, coletando dados sobre o salário e número de filhos. A prefeitura deseja saber:

   i. média do salário da população;
   ii. média do número de filhos;
   iii. maior salário;
   iv. percentual de pessoas com salário até R$100,00.

   Desenvolver um programa que calcule e imprima as informações acima. O final da leitura de dados se dará com a entrada de um salário negativo.

2. **Comando `switch`**

   Reescreva o programa do Exercício 1 da Parte 1 para, após apresentar o menu ao usuário, ler um número inteiro e imprimir na tela a opção de menu correspondente. Usar o comando `switch` para determinar a opção escolhida. Tratar devidamente a leitura de valores inválidos!

Parte 3 – Funções

Para todos os exercícios a seguir, faça uma função `main` para testar o procedimento ou função desenvolvido.

1. **Função**

   Desenvolver uma função que recebe como parâmetros o dia e o mês do ano e imprime a estação do ano correspondente.

2. **Funções auxiliares**

   Desenvolver uma função que receba como parâmetro um valor n inteiro e positivo e que calcule e retorne a seguinte soma:

   $$S = n + \frac{n-1}{2!} + \frac{n-2}{3!} + \frac{n-3}{4!} + \cdots + \frac{1}{n!} = \sum_{i=0}^{n} \frac{n-i}{(i+1)!}$$

   Faça uma função auxiliar para calcular o fatorial e use-a na função que calcula o somatório acima.

3. **Função booleana**

   Desenvolver uma função que receba como parâmetro um número inteiro n e determine se ele é divisível por 3 e 5 ao mesmo tempo. Utilize o tipo de dados `bool` do C++.

Parte 4 – Arrays

1. **Impressão de tabela.**

   Desenvolver um programa que leia 20 números inteiros e imprima a média dos números lidos ao lado de cada número.

   Exemplo de saída:

```
Valor | Media
_____________

  7   --  40
  15  --  40
  73  --  40
      ...
  60  --  40
```

2. **Vetores como parâmetros.**

   Desenvolver um conjunto de procedimentos e funções para:

   i. inicializar um vetor com zeros;
   ii. inserir um valor em uma posição específica do vetor
   iii. excluir um valor de uma posição específica do vetor
   iv. retornar o maior elemento do vetor
   v. calcular e imprimir a média dos elementos do vetor
   vi. verificar e retornar se um determinado valor está no vetor

   Faça uma função main que utilize todos os procedimentos e funções acima para um vetor de 50 posições inteiras.

3. **Matrizes.**

   Desenvolver um conjunto de procedimentos e funções para:

   i. inicializar a matriz com a identidade;
   ii. somar duas matrizes e guardar o resultado em uma terceira matriz
   iii. multiplicar duas matrizes e guardar o resultado em uma terceira matriz
   iv. transpor uma matriz e guardar o resultado em uma outra matriz

   Faça uma função main que utilize todos os procedimentos e funções acima para uma matriz 10×10 de elementos reais.

Parte 5 – Strings

1. **Strings C-*style***

   Desenvolver uma função que recebe como parâmetro uma string no formato do C (ou seja, na forma de um vetor de caracteres) e retorne essa mesma string no formato do C++. Não utilize nenhuma função pré-definida do tipo string de C++ nem funções da biblioteca `string.h`. Use apenas manipulação básica de vetores e o operador de concatenação em C++.

2. **Particionamento de strings**

   Desenvolver um programa que leia uma string e um caracter. O programa deve quebrar a string em cada ocorrência deste caracter, sem incluir o caracter, e imprimir as strings resultantes.

   Exemplo:

```
String: Fulana comeu dois bolinhos.

Caracter: <espaço>
Strings resultantes:
Fulana
comeu
dois
bolinhos.

Caracter: i
Strings resultantes:
Fulana comeu do
s bol
nhos.
```

3. **Particionamento de strings com vetores**

   Modifique o exercício anterior para guardar cada string resultante em um vetor de strings. Em seguida, imprima uma tabela, contendo a string e o tamanho da string (quantidade de caracteres) ao lado.

   Exemplo:

```
   String   |   Tamanho
______________________

   Fulana   --     6
   comeu    --     5
    dois    --     4
  bolinhos  --     8
```

4. **Substituição de caracteres**

   Desenvolver um programa que leia uma string e substitua todas as vogais pela vogal subsequente no alfabeto (isto é, a –> e, e –> i, i –> o, o –> u, u –> a).

5. **Minha `string.h`**

   Desenvolva novas versões das funções de string da linguagem C contidas na `string.h` (`strlen`, `strcat`, `strcpy` etc.) para utilização do tipo string de C++.

   SUGESTÃO: Você pode salvar os protótipos das funções em um arquivo .h (por exemplo, `mystring.h`) e incluir este arquivo no seu `main.cpp`. Basta ir no menu New/File do Code::Blocks, criar um C/C++ header e adicioná-lo ao projeto.