Nome: Gabriel Victorino

Exercicios

1) Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

2) Texto

Descrição gerada automaticamente com confiança média

3)

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

4)

Texto

Descrição gerada automaticamente

5)

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

6)Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Estrutura de Decisão

7)

Texto

Descrição gerada automaticamente

8)

Texto

Descrição gerada automaticamente

9)

Texto

Descrição gerada automaticamente

10)Texto

Descrição gerada automaticamente

11)

Texto

Descrição gerada automaticamente

12)

Texto

Descrição gerada automaticamente

13)

#include <iostream>

#include <string.h>

using namespace std;

//validador de cpf

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char\*\* argv) {

int soma=0, resultado = 0, digito10, digito11, digito=0;

char cpf[11];

cout << "Informe o cpf (Sem espaços e pontos). \n\n";

cin >> cpf;

if(strlen(cpf) != 11){

cout << "Cpf nao possui 11 numeros.\n";

return 0;

}else if(

(strcmp(cpf,"00000000000") == 0) || (strcmp(cpf,"11111111111") == 0) || (strcmp(cpf,"22222222222") == 0) ||

(strcmp(cpf,"33333333333") == 0) || (strcmp(cpf,"44444444444") == 0) || (strcmp(cpf,"55555555555") == 0) ||

(strcmp(cpf,"66666666666") == 0) || (strcmp(cpf,"77777777777") == 0) || (strcmp(cpf,"88888888888") == 0) ||

(strcmp(cpf,"99999999999") == 0)

){

cout << "Todos os numeros repetidos.\n\n";

return 0;

}else{

//digito verificador 1

for(int i = 0; i < 9; i++){

digito = cpf[i]-'0';

// cout << digito << endl;

soma+= digito \* (10 - i);

// cout << soma << endl;

}

digito10 = (soma \* 10) % 11;

if(digito10 == 10 || digito10 == 11){

digito10 = 0;

}

//digito verificador 2

soma = 0;

for(int i = 0; i < 10; i++){

digito = cpf[i]-'0';

// cout << digito << endl;

soma+= digito \* (11 - i);

// cout << soma << endl;

}

digito11 = (soma \* 10) % 11;

if(digito11 == 10 || digito11 == 11){

digito10 = 0;

}

if(digito10 == cpf[9]-'0' && digito11 == cpf[10]-'0'){

cout << "CPF VALIDO!" << endl;

}else{

cout << "CPF INVALIDO" << endl;

}

}

return 0;

}

14)

#include <iostream>

#include <vector>

#include <algorithm>

#include <iterator> // Inclua este cabeçalho para usar ostream\_iterator

using namespace std;

int main() {

int NUM\_MAX = 0;

cout << "Informe um numero maximo: \n\n";

cin >> NUM\_MAX;

vector<int> vetor;

//popula o vetor

for(int i = 1; i <= NUM\_MAX; i++){

vetor.push\_back(i);

}

cout << endl;

// // Removendo o primeiro ultimo elemento

// vetor.erase(vetor.begin());

// vetor.pop\_back();

copy(vetor.begin(), vetor.end(), ostream\_iterator<int>(cout, " "));

cout << std::endl;

while(vetor.size() > 1){

// Removendo o primeiro ultimo elemento

vetor.erase(vetor.begin());

vetor.pop\_back();

copy(vetor.begin(), vetor.end(), ostream\_iterator<int>(cout, " "));

cout << std::endl;

}

return 0;

}

15)