Ciência da Computação – 2º Período

Benaia Rodrigues

José Henrique

Letícia Emanuela

Trabalho Prático 01

Ipatinga – MG, 25 de setembro de 2016

Benaia Rodrigues

José Henrique

Letícia Emanuela

Trabalho Prático 01

Trabalho desenvolvido durante a disciplina de Introdução à Organização de Arquivos e Dados, como parte da avaliação referente a 1ª etapa.

Professor: Thiago de Medeiros Gualberto

Ipatinga – MG, 25 de setembro de 2016

Introdução

No trabalho a seguir vamos abordar a parte teórica das questões “mega sena” e “calculadora romana”, onde falaremos das dificuldades do grupo e como conseguimos soluciona-las, e iremos explicar um pouco cada código (algoritmo).

1. Parte I – Mega Sena

Foi desenvolvido um algoritmo (linguagem C++) pelo grupo que simula um programa de sorteio, onde o usuário informe o valor inicial, final e a quantidade de números a serem sorteados. Gerando assim várias possibilidades de sorteio sem repetições de números.

1.1 Programa

O programa deve ler três números A, B e X, e em seguida sortear X números inteiros sem repetição no intervalo [A,B], ou seja, o menor número possível é A e o maior é B.

1.1.1. Código

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
#include<iostream>
#include<time.h>
#include<windows.h>

using namespace std;

#define valido   1
#define invalido 0

int main()
{
   int a,b,x;
   int vet['x'];
   int teste;

   cout<<"\n\t\t\t\t MEGA SENA \n\n";
   cout<<"  Informe o valor inicial: ";
   cin>>a;
   cout<<"\n  Informe o valor final: ";
   cin>>b;
   cout<<"\n  Informe a  quantidade de numeros para o sorteio: ";
   cin>>x;

   system("cls");
```

Bibliotecas essenciais para o funcionamento do código.

Constantes usadas para não deixar os números se repetirem.

```
  while((b<x) || (a>=b)) //Condições de erro...
  {
        cout<<"\n\t\t SORTEIO IMPOSSIVEL, TENTE OUTRA VEZ!!!\n\n";
        system("pause");
        system("cls");

        cout<<"\n\t\t\t\t MEGA SENA \n\n";
        cout<<"  Informe o valor inicial: ";
        cin>>a;
        cout<<"\n  Informe o valor final: ";
        cin>>b;
        cout<<"\n  Informe a  quantidade de numeros para o sorteio: ";
        cin>>x;
        system("cls");
  }


  cout<<"\n\t\t\t\t MEGA SENA \n\n";
  cout<<"\n  Inicio: "<<a;
  cout<<"\n\n  Fim: "<<b;
  cout<<"\n\n  Quantidade: "<<x;

  cout<<"\n\n\n\n\t\t\t\t  SORTEIO \n\n";

  srand(time(NULL));

   for(int i=0;i<x;i++)       //Exibe os numeros sorteados
   {
        do                    // Atribui os numeros sorteados em um vetor
        {
             vet[i]= a + rand() % (b-a);
             teste=valido;

            for(int j=0; j<i;j++)          //Compara os numeros para não repetir nenhum
            {
                 if(vet[i]==vet[j])
                   {
                      teste=invalido;
                   }
             }

         } while(teste==invalido);

                  cout<<"  "<<vet[i];
                  Sleep(1000);            //Faz os numeros aparecerem lentamente
    }

getch();
return 0;
}
```

1.1.2 Exemplos

Inicio: 9
Fim: 15
Quantidade: 6
Sorteio: 11 9 13 14 10 12

Inicio: 1
Fim: 100
Quantidade: 6
Sorteio: 62 51 80 76 90 15

Inicio: 1
Fim: 50
Quantidade: 60
Sorteio: Sorteio impossível, tente outra vez!!!

1.2 Dificuldades

O nível de dificuldade da questão foi bem baixo, o grupo não apresentou muitas dificuldades, a maior dificuldade apresentada foi fazer os números não repetirem e nem ser menor que o valor inicial.

1.3 Soluções

Após várias tentativas o grupo resolveu recorrer a internet para buscar um método para o número não ser menor que o inicial, onde achamos a seguinte formula "a + rand() % (b-a)" que seria "valor inicial + números aleatórios % (valor final - valor inicial).

Para resolver o erro dos números repetidos nós usamos duas constantes "valido" e "invalido", onde valido = 1 (verdadeiro) e invalido = 0 (falso), um laço do-while para atribuir os valores aleatórios ao vetor e um laço for (dentro do do-while) para comparar os valores, caso os números fossem iguais a variável teste recebia invalido (ou seja, teste = 0) fazendo um novo sorteio para a posição do vetor.

1.4 Acréscimos ao código

Foi acrescentado ao código uma condição de erro, para informar o usuário a frase “SORTEIO IMPOSSIVEL, TENTE OUTRA VEZ!!!”, quando for informado um valor inicial (a) maior que o valor final (b) ou quando a quantidade de números sorteados (x) for maior que o valor final.

2. Parte II – Calculadora Romana

Foi desenvolvido um algoritmo (linguagem C++) pelo grupo que é capaz de fazer contas (+ - * /), porem as contas não serão efetuadas com números decimais e sim com números romanos.

2.1 Programa

Desenvolver um programa que leia 2 números em algarismo romano, o sinal de operação e, após este procedimento, realizar o cálculo da operação sobre os números romanos e mostrar o resultado em número romano na tela.

2.1.1 Código

```
#include<iostream>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<string>
using namespace std;

int main()
{
	string nr3, nr4, op;
	int nd1=0, nd2=0, n1=0, n2=0, r, rf=0, aux, aux2, u=0;
	char result[99];

	cout<<"Digite um numero romano: "<<endl;
	cin>>nr3;
	cout<<"\nInforme o operador: "<<endl;
	cin>>op;
	cout<<"\nDigite um numero romano: "<<endl;
```

```
cin>>nr4;

for (int i = 0; i < nr3.length(); i++)
{
	if(nr3[i]=='I' || nr3[i]=='i')
	{
		nd1=nd1+1;
		if(nr3[i+1]=='V' || nr3[i+1]=='v')
		{
			nd1=nd1+3;
			i++;
		}

		else
		if(nr3[i+1]=='X' || nr3[i+1]=='x')
		{
			nd1=nd1+8;
			i++;
		}

	}

	else
	if(nr3[i]=='V' || nr3[i]=='v')
	{
		nd1=nd1+5;
	}

	else
	if(nr3[i]=='X' || nr3[i]=='x')
```

```
{
        nd1=nd1+10;
        if(nr3[i+1]=='L' || nr3[i+1]=='l')
        {
                nd1=nd1+30;
                i++;
        }

        else
        if(nr3[i+1]=='C' || nr3[i+1]=='c')
        {
                nd1=nd1+80;
                i++;
        }
}

else
if(nr3[i]=='L' || nr3[i]=='l')
{
        nd1=nd1+50;
}

else
if(nr3[i]=='C' || nr3[i]=='c')
{
        nd1=nd1+100;
        if(nr3[i+1]=='D' || nr3[i+1]=='d')
        {
                nd1=nd1+300;
                i++;
```

```
                }

                else
                if(nr3[i+1]=='M' || nr3[i+1]=='m')
                {
                        nd1=nd1+800;
                        i++;
                }
        }

        else
        if(nr3[i]=='D' || nr3[i]=='d')
        {
                nd1=nd1+500;
        }

        else
        if(nr3[i]=='M' || nr3[i]=='m')
        {
                nd1=nd1+1000;
        }
}

for (int i = 0; i < nr4.length(); i++)
{
        if(nr4[i]=='I' || nr4[i]=='i')
        {
                nd2=nd2+1;
                if(nr4[i+1]=='V' || nr4[i+1]=='v')
                {
```

```
                nd2=nd2+3;
                i++;
        }

        else
        if(nr4[i+1]=='X' || nr4[i+1]=='x')
        {
                nd2=nd2+8;
                i++;
        }
}

else
if(nr4[i]=='V' || nr4[i]=='v')
{
        nd2=nd2+5;
}

else
if(nr4[i]=='X' || nr4[i]=='x')
{
        nd2=nd2+10;
        if(nr4[i+1]=='L' || nr4[i+1]=='l')
        {
                nd2=nd2+30;
                i++;
        }

        else
        if(nr4[i+1]=='C' || nr4[i+1]=='c')
```

```
        {
                nd2=nd2+80;
                i++;
        }
}

else
if(nr4[i]=='L' || nr4[i]=='l')
{
        nd2=nd2+50;
}

else
if(nr4[i]=='C' || nr4[i]=='c')
{
        nd2=nd2+100;
        if(nr4[i+1]=='D' || nr4[i+1]=='d')
        {
                nd2=nd2+300;
                i++;
        }

        else
        if(nr4[i+1]=='M' || nr4[i+1]=='m')
        {
                nd2=nd2+800;
                i++;
        }
}
```

```
        else
        if(nr4[i]=='D' || nr4[i+1]=='d')
        {
                nd2=nd2+500;
        }

        else
        if(nr4[i]=='M' || nr4[i+1]=='m')
        {
                nd2=nd2+1000;
        }
}


if(op[0]=='+')
{
        n1=nd1+nd2;
}

else
if(op[0]=='-')
{
        n1=nd1-nd2;
}

else
if(op[0]=='*')
{
        n1=nd1*nd2;
}
```

```
        else
        if(op[0]=='/')
        {
                n1=nd1/nd2;
        }

        else
        if(op[0]!='+' && op[0]!='-' && op[0]!='*' && op[0]!='/')
        {
                cout<<"Operador Invalido\n";
                system("pause");
        }

        aux=n1;

        if (n1 == 0) n2 = 1;

else
        {
            while (n1 != 0)
       {
          n2 = n2 + 1;
          n1 = n1 / 10;
       }
        }

        do
        {
                if(n2==1)r=1;
```

```
if(n2==2)r=10;
if(n2==3)r=100;
if(n2==4)r=1000;
if(n2==5)r=10000;
if(n2==6)r=100000;
if(n2==7)r=1000000;
if(n2==8)r=10000000;
if(n2==9)r=100000000;
if(n2==10)r=1000000000;

rf=(aux%r);
n2=n2-1;
aux2=aux/r;

if(aux>=1000)
{
        do
        {

                strcat(result,"M");
                aux2=aux2-1;
                u++;

        }while(aux2>0);

        aux=aux-(1000*u);
}

if(aux2!=0)
if(aux>=900)
```

```
{
        strcat(result,"CM");
        aux=aux-900;
        aux2=aux2-9;
}

if(aux2!=0)
if(aux>=500)
{
        strcat(result,"D");
        aux=aux-500;
        aux2=aux2-5;
}

if(aux2!=0)
if(aux>=400)
{
        strcat(result,"CD");
        aux=aux-400;
        aux2=aux2-4;
}

if(aux2!=0)
if(aux>=100)
{
        u=0;
        do{

                strcat(result,"C");
                aux2=aux2-1;
```

```
                u++;

        }while(aux2>0);
        aux=aux-(100*u);
}

if(aux2!=0)
if(aux>=90)
{
        strcat(result,"XC");
        aux=aux-90;
        aux2=aux2-9;
}

if(aux2!=0)
if(aux>=50)
{
        strcat(result,"L");
        aux=aux-50;
        aux2=aux2-5;
}

if(aux2!=0)
if(aux>=40)
{
        strcat(result,"XL");
        aux=aux-40;
        aux2=aux2-4;
}
```

```
if(aux2!=0)
if(aux>=10)
{

        u=0;
        do{

                strcat(result,"X");
                aux2=aux2-1;
                u++;

        }while(aux2>0);
        aux=aux-(10*u);
}

if(aux2!=0)
if(aux>=9)
{
        strcat(result,"IX");
        aux=aux-9;
        aux2=aux2-9;
}

if(aux2!=0)
if(aux>=5)
{
        strcat(result,"V");
        aux=aux-5;
        aux2=aux2-5;
}
```

```
                if(aux2!=0)
                if(aux>=4)
                {
                        strcat(result,"IV");
                        aux=aux-4;
                        aux2=aux2-4;
                }

                if(aux2!=0)
                if(aux>=1)
                {
                        u=0;
                        do{

                                strcat(result,"I");
                                aux2=aux2-1;
                                u++;
                        }while(aux2>0);
                        aux=aux-(1*u);
                }

        }while(aux > 0);

        cout<<"\nO Resultado Final e: "<<result<<endl;

system("pause");
return 0 ;
}
```

2.1.2 Exemplos

C + C = CC

LX + X = LXX

MMIII – XVII = MCMLXXXV

MMMDCCCLXXXVII – MMMCCCXXXIII = MDLV

MCL / II = DLXXV

L * II = C

2.2 Dificuldades

As maiores dificuldades apresentadas foi fazer as operações com os números romanos, ler mais de um caracter (ex: XXX), e como organizar as letras de acordo com seu valor.

2.3 Soluções

Para ler mais de um caracter foi usado uma variável declarada como string, pois automaticamente ela já armazena vários caracteres, usamos um for para percorrer cada posição da string para ler cada caracter e ir calculando seu valor final.

Depois de ter o valor final definido transformamos todos os números para decimal primeiro e efetuamos a operação e depois voltamos ele para algarismo romano. Usamos um vetor para armazenar o resultado final já transformado em algarismos romanos, e fizemos uma função para determinar o tamanho do vetor.

2.4 Acréscimos ao código

Foi acrescentado ao código uma condição de erro, para informar o usuário que não é possível realizar a operação quando for informado um valor no lugar do operador (+ - * /).

Bibliografia

Parte I – Mega Sena

http://www.scriptbrasil.com.br/forum/topic/116192-armazenar-valores-aleat%C3%B3rios-n%C3%A3o-repetidos-num-vetor/

http://pt.stackoverflow.com/questions/39215/faixa-de-n%C3%BAmeros-aleat%C3%B3rios-em-c

Parte II – Calculadora Romana

http://numeracaoromana.babuo.com/numeros-romanos-de-1-a-5000