Trabalho 2

Alunos: Fernanda Maria de Souza e Nikolas Jensen.

Disciplina: Algoritmos.

/\* Arquivo de texto input:

0 63 210 190 0 190

63 0 160 150 95 0

210 160 0 10 0 0

190 150 10 0 0 0

0 95 0 0 0 80

190 0 0 0 80 0

\*/

/\* Arquivo de texto cd:

1 8 0 4 58 9 0 0 3 7 4 5

6 78 9 3 4 5 6 1 2 3 8 9

8 7 9 2 3 4 5 8 4 5 6 8

1 2 38 4 5 6 78 8 9 96 5 7

1 2 38 4 5 6 78 8 9 96 5 7

8 7 9 2 3 4 5 8 4 5 6 8

8 7 9 2 3 4 5 8 4 5 6 8

8 7 9 2 3 4 5 8 4 5 6 8

8 7 9 2 3 4 5 8 4 5 6 8

1 8 0 4 58 9 0 0 3 7 4 5

1 8 0 4 58 9 0 0 3 7 4 5

1 8 0 4 58 9 0 0 3 7 4 5

\*/

**1- Faça um programa que leia a matriz e o vetor e calcule a distância percorrida durante a viagem.**

#include <stdlib.h>

#include <stdio.h>

int main () {

int m[6][6],i,j,s=0,x;

FILE \*input = fopen ("input.txt", "r");

if (input==NULL){

printf("\nNao deu certo");

return -1;

}

printf("Leu o arquivo\n");

while ( !feof(input)){

for (i=0;i<6;i++) {

for (j=0;j<6;j++) {

fscanf (input, "%d", &m[i][j]);

}

}

}

int v[5] = {m[2][3],m[3][1],m[1][4],m[4][5],m[5][0]};

for(x=0;x<=5;x++){

s = v[x] + s;

}

printf ("A distancia percorrida durante a viagem foi de %d km.", s);

return 0;

}

**2- Considere a matriz da questão anterior como entrada. Faça um programa que mostre o número e o nome das cidades, e então o usuário pode digitar o número das cidades para compor uma rota válida. O programa deve calcular a distância da rota válida que o usuário fornecer.**

#include <stdlib.h>

#include <stdio.h>

int main () {

int m[6][6],i,j,s=0,x,u,v[6];

FILE \*input = fopen ("input.txt", "r");

if (input==NULL){

printf("\nNao deu certo");

return -1;

}

printf("Leu o arquivo\n");

while ( !feof(input)){

for (i=0;i<6;i++) {

for (j=0;j<6;j++) {

fscanf (input, "%d", &m[i][j]);

}

}

}

for(i=0;i<6;i++){

printf("Entre com um numero de cidade\n");

printf("0: Carceres\t1: B. do Bugres\t2: Cuiaba\t3: V. Grande\t4: Tangara\t5: P. Lacerda\n");

scanf("%d",&v[i]);

}

int o[5] = {m[v[0]][v[1]],m[v[1]][v[2]],m[v[2]][v[3]],m[v[3]][v[4]],m[v[4]][v[5]]};

for(x=0;x<=5;x++){

s = o[x] + s;

}

printf ("%d", s);

return 0;

}

**3- Leia uma matriz quadrada qualquer e crie um programa para calcular a média aritmética dos elementos hachurados da figura 1, letra a.**

#include <stdlib.h>

#include <stdio.h>

int main () {

int M[12][12],i,j,soma,cont,media;

FILE \*cd = fopen ("cd.txt", "r");

if (cd==NULL){

printf("\nNao deu certo");

return -1;

}

while ( !feof(cd)){

for (i=0;i<12;i++) {

for (j=0;j<12;j++) {

fscanf (cd, "%d", &M[i][j]);

}

}

}

soma = 0;

cont = 0;

for (i=2;i<=6;i++) {

for (j=i+1;j<12-i;j++) {

soma += M[i][j];

soma += M[13-i][j];

cont++;

}

}

media = soma/cont;

printf ("A media e: %d", media);

}

**4- Com a mesma matriz que você leu no item 3, crie um programa para achar o maior elemento da linha onde se encontra o menor elemento da área hachurada na letra b da Figura 1.**

#include <stdlib.h>

#include <stdio.h>

int main () {

int M[12][12],i,j,menor,maior,auxi,auxj;

FILE \*cd = fopen ("cd.txt", "r");

if (cd==NULL){

printf("\nO arquivo nao foi lido.");

return -1;

}

while ( !feof(cd)){

for (i=0;i<12;i++) {

for (j=0;j<12;j++) {

fscanf (cd, "%d", &M[i][j]);

}

}

}

menor = 0;

for (i=0;i<11;i++) {

for (j=0;j<12-i;j++) {

if (M[i][j]<menor) {

menor = M[j][i];

auxi = i;

}

}

}

maior = 0;

for (i=0;i<11;i++) {

for (j=0;j<12-i;j++) {

if (M[auxi][j]>maior) {

maior = M[i][j];

auxj = j;

}

}

}

printf ("O maior elemento da linha onde se encontra o menor elemento da area hachurada e: %d", maior);

return 0;

}

**4- Faça um programa que leia uma matriz 12x12. Calcule a soma da área hachurada na letra a da figura 2.**

#include <stdlib.h>

#include <stdio.h>

int main () {

int M[12][12], i, j, soma;

FILE \*cd = fopen ("cd.txt", "r");

if (cd==NULL){

printf("\nO arquivo nao foi lido.");

return -1;

}

while ( !feof(cd)){

for (i=0;i<12;i++) {

for (j=0;j<12;j++) {

fscanf (cd, "%d", &M[i][j]);

}

}

}

for (i=0;i<11;i++) {

for (j=0;j<12-i;j++) {

soma += M[i][j];

}

}

printf ("A soma da area hachurada e: %d", soma);

return 0;

}

**5- Com a mesma matriz que você leu no item 4, faça um programa que escreva o maior elemento da área hachurada na letra b. Use leitura da matriz 12x12 através de arquivo.**

#include <stdlib.h>

#include <stdio.h>

int main () {

int M[12][12],i,j,maior;

FILE \*cd = fopen ("cd.txt", "r");

if (cd==NULL){

printf("\nO arquivo nao foi lido.");

return -1;

}

while ( !feof(cd)){

for (i=0;i<12;i++) {

for (j=0;j<12;j++) {

fscanf (cd, "%d", &M[i][j]);

}

}

}

maior = 0;

for (i=0;i<6;i++) {

for (j=i;j<13-i;j++) {

if (M[i][j]>maior) {

maior = M[j][i];

}

if (M[j][13-i]>maior) {

maior = M[j][13-i];

}

}

}

printf ("O maior elemento da area hachurada e: %d", maior);

return 0;

}

**6- Escreva um programa que leia n e imprima todos os inteiros compreendidos entre 1 e n que são "perfeitos", isto é, iguais à soma dos seus divisores próprios; um divisor d de n diz-se próprio se é inferior a n. Por exemplo, 6 é perfeito porque 6=1+2+3. O seu programa deve incluir uma função "int somad(int n)" que retorna a soma dos divisores próprios de n.**

#include <math.h>

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n,r,i,j,k=0,l;

printf ("Escreva um numero.");

scanf("%d",&n);

for(i=2;i<=n;i++){

for(j=1;j<i;j++){

if(j!=0){

if(i%j==0){

k += j;

}

}

}

if(k==i){

l++;

printf("Numero Perfeito: %d\n",i);

}

k=0;

}

printf("Numero de Perfeitos na sequencia: %d\n",l);

r = somad(n);

printf("Soma dos divisores de n: %d\n",r);

return 0;

}

int somad(int n){

int soma,i,j,k=0;

for(i=0;i<n;i++){

if(i!=0){

if(n%i==0){

k += i;

}

}

}

soma = k;

return(soma);

}

**7- Escreva um programa que diga de um N dado pelo usuário pertence à série de Fibonacci. A série de fibonacci deve ser calculada por uma função como “int fibonacci ( int N )”.**

#include <stdio.h>

#include <stdlib.h>

main()

{

int N,i;

printf("Digite o numero para verificacao: ");

scanf("%d", &N);

if (fibonacci(N)) {

printf ("O numero %d e um numero de Fibonacci.", N);

}

else {

printf ("O numero %d nao e um numero de Fibonacci", N);

}

}

int fibonacci(int N)

{

int i, x = 0, y = 1, soma, cont=0;

for (i=0; i<=N; i++)

{

soma = x + y;

x = y;

y = soma;

if(soma==N){

cont++;

}

}

if (cont == 1) {

return 1;

}

else {

return 0;

}

}