1. Palindrome

* Codes :

#include<stdio.h>

#include<string.h>

//author : Evelyn Tjitrodjojo

int main(){

int a,b,c,i,j,k,h,t,g=0;

char d;

scanf("%d %d",&a,&b);

char z[100001];

scanf("%s",&z);

t=strlen(z);

if(t==a)

{

h=0;

for(j=0,k=t-1;j<=((t-1)/2);j++,k--)

{

if(z[j]!=z[k])

{

h++;

}

}

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

{

scanf("%d %c",&c,&d);

if(z[c]!=z[t-c-1]){

z[c]=d;

if(z[c]==z[t-1-c])

{

h--;

}

}

else if(z[c]==z[t-c-1])

{

z[c]=d;

if(z[c]!=z[t-1-c])

{

h++;

}

}

if(h>0) printf("BUKAN\n");

else printf("YA\n");

}

}

return 0;

}

* Penjelasan :
* Input :
* Output:

1. Bolak-balik Matriks

* Codes :

#include<stdio.h>

//author : Evelyn Tjitrodjojo

//translator : M.Hilmi Ramadhan

int main(){

int n,i,j;

char r,v,z=0,y=0;

scanf("%d",&n); //input ordo

int x[n][n];

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

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

scanf("%d",&x[i][j]); //input angka matriks

}

}

int m[n][n];

while(y!='z'){

scanf("\n%c",&y); //input char untuk rotasi (right dan flip

if(y=='r'){

scanf("\n%c",&z); //r untuk right

if(z=='r'){

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

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

m[i][j]=x[n-j-1][i]; //transpose right

printf("%d ",m[i][j]);

}

printf("\n");

}

}

else if(z=='l'){ //rotasi ke kiri

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

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

m[i][j]=x[j][n-i-1];

printf("%d ",x[j][n-i-1]);

}

printf("\n");

}

}

else if(z=='z') break; //input z untuk break

else {

printf("YANG BENER WOY!\n");

}

}

else if(y=='f'){

scanf("\n%c",&z); //input flip

if(z=='h'){ //input h untuk horizontal

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

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

printf("%d ",x[n-i-1][j]);

m[i][j]=x[n-i-1][j];

}

printf("\n");

}

}

else if(z=='v'){ //input flip vertical

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

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

printf("%d ",x[i][n-j-1]);

m[i][j]=x[i][n-j-1];

}

printf("\n");

}

}

else if(z=='z') break; //input z untuk break

else {

printf("YANG BENER WOY!\n");

}

}

else if(y=='z') break;

else {

printf("YANG BENER WOY!\n");

}

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

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

x[i][j]=m[i][j];

}

}

}

return 0;

}

* Penjelasan :

1. Pada dasarnya program ini mirip dengan matriks transpose, namun perbedaanya ialah jika matriks transpose bergerak clockwise.
2. Matriks ini dapat bergerak pindah tempat seperti transpose dengan arah tertentu missal ke kanan, kiri, atau bahkan flip secara vertical dan horizontal.
3. Prinsip kerja dari program ini ialah menggunakan for loop pada umumnya untuk menentukan baris dan kolom dari inputan user, setelah itu menggunakan konsep transpose namun dengan arah yang lebih variasi.

* Input : Diminta untuk memasukkan ordo matriks yang stara nxn, kemudian menginputkan sesuai dengan ordo tiap baris dan kolom.
* Output: user dapat melakukan pemutaran left, right, dan pada flip dapat pemutaran terhadap vertical dan horizontal pada matriks inputan.

1. Nilai Rata-Rata

* Codes :

/\*Menghitung nilai rata-rata

dari n buah bilangan yang diinputkan \*/

#include<stdio.h>

//author : Evelyn Tjitrodjojo

//translator : M. Hilmi Ramadhan

int main () {

int n, a, b, c, rata\_rata, hasil;

rata\_rata=0;

printf("Tuliskan Nilai n = ");

scanf("%d", &n); //input jumlah array

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

scanf("%d", &b); //input data array

rata\_rata=rata\_rata+b; //jumlah data array

}

hasil=rata\_rata/n; //jumlah dibagi dengan total array

printf("%d", hasil); //print rata-rata

}

* Penjelasan :

1. Program ini bertujuan untuk menghitung sejumlah angka inputan user dan mencari nilai rata-ratanya.
2. Cara kerja program ini ialah dengan meminta inputan sejumlah batas array terlebih dahulu kemudian diikuti inputan untuk mengisi sederet angka kepada array yang dimaksud.
3. Selanjutnya, perulangan for loop dengan batas maksimal adalah inputan batas array useryang bertujuan untuk menjumlah isi data dari array.
4. Setelah kondisi for loop memenuhi, maka dapat selesai perulangan dan masuk ke pembagian dri jumlah yg telah dihitung dengan banyaknya batas maksimal array yg ditentukan.
5. Terakhir tampilkan hasil ke layar pengguna.

* Input : Meminta berapa banyak tempat array yg akan diisi, kemudian diberikan sebuah deret angka dengan batas yg telah diisi tadi. Missal 6, kemudian inputan lanjutan ialah 5 4 3 5 2 1.
* Output: Menghasilkan perhitungan rata-rata atau dalam matematika dikenal sebagai mean. Misal hasil rata-rata dri inputan diatas adalah 3.

1. Persamaan Kuadrat

* Codes :

#include<stdio.h>

#include<math.h>

//author : Evelyn Tjitrodjojo

//translator : M. Hilmi Ramadhan

int main(){

int a,b,c;

float x,akar1,akar2;

printf("Persamaan kuadrat\n");

printf("a = ");

scanf("%d",&a); //input a

printf("\nb = ");

scanf("%d",&b); //input b

printf("\nc = ");

scanf("%d",&c); //input c

x= b\*b-4\*a\*c;

if (x>=0){

akar1 = (-b+sqrt(b\*b-4\*a\*c))/2\*a; //menggunakan rumus persamaan kuadrat

akar2 = (-b-sqrt(b\*b-4\*a\*c))/2\*a; //menggunakan rumus persamaan kuadrat

printf("\nakar 1 = %f\n",akar1); //hasil akar pertama

printf("akar 2 = %f\n",akar2); //hasil akar kedua

}

else{

printf("angka imajiner"); //selain itu angka imajiner

}

return 0;

}

* Penjelasan :

1. Program ini bertujuan untuk menghitung persamaan akar kuadrat.
2. Dimana meminta inputan a, b, dan c. Variabel-variabel ini nantinya akan digunakan mencari dari Diskriminan terlebih dahulu.
3. Selanjutnya setelah dihitung dapat nilai D, maka jika D >= 0 maka dapat dihitung akar1 dan akar 2 nya melalui rumus abc plus minus.
4. Sebaliknya jika D<<=0, maka langsung menghasilkan angka imajiner.

* Input : angka untuk mengisi variable a,b, dan c. Yang nantinya akan digunakan sebagai mencari diskriminan terlebih dahulu, jika diskriminanya lebih besar sama dengan 0 maka dapat dicari akar akar persamaan kuadratnya.
* Output: Hasil dari akar-akar persamaan kuadratnya, baik itu dapat ditentukan maka akan menghasilkan akar1 dan akar2. Tetapi jika D<=0 maka langsung menghasilkan angka imajiner.

1. Fibonacci

* Codes :

#include<stdio.h>

//author : Evelyn Tjitrodjojo

//translator : M. Hilmi Ramadhan

int main(){

int a=0,b=1,c,n;

c=a+b;

printf("Bilangan Fibonacci\n");

printf("Masukkan angka: ");

scanf("%d",&n); //input angka fibonacci

while(n!=0){

printf("%d ",b);

a=b; //swap

b=c;

c=a+b; //jumlah a dan b

n-=1; //n dikurang 1

}

return 0;

}

* Penjelasan : ([*https://rumuspintar.com/fibonacci/*](https://rumuspintar.com/fibonacci/))

1. Pada dasarnya ini adalah implementasi dari bil.fibonacci yang di programkan ke C dengan menggunakan perulangan while loop.
2. Cara kerja dri program ini ialah meminta inputan user sbg variable n yang nantinya n akan masuk sbg parameter dari perulangan while loop atau batasnya sampai tidak sama dengan n maka loop selesai.
3. Pada perulangan, terjadi proses swapping untuk menghasilkan bilangan fibo hingga sesuai banyaknya n test case yang diminta oleh user.
4. Ketika kondisi sudah memenuhi maka dapat di layar user.

* Input : Angka test case untuk berapa banyak bilangan Fibonacci yang nantinya akan ditampilkan ke layer. Misal 3
* Output: Menampilkan hasil bilangan Fibonacci dengan banyak sesuai inputan user. Misal 1 1 2.

1. Bilangan Faktorial

* Codes :

#include<stdio.h>

//author : Evelyn Tjitrodjojo

//translator : M.Hilmi Ramadhan

int main(){

int result,n,x;

printf("Bilangan faktorial = ");

scanf("%d",&n); //input angka yang difaktorial

x=n;

result=1;

while(n>1){

result=result\*n; //angka dikali angka-1

n--;

}

printf("N = %d\n",x); //print angka faktorial

printf("N! = %d",result); //print hasil

return 0;

}

* Penjelasan :

Program ini bertujuan utk menghitung factorial dri sebuah integer inputan test case.

Menggunakan while loop dengan decrement persis seperti rumus dari factorial asli missal 4! = 4.3.2.1.

Perulangan terus berlangsung hingga n berubah menjadi 1>1 baru selesai dan dapat di tampilkan ke layer.

* Input : Sebuah angka test case yg akan dihitung faktorialnya missal 4.
* Output: merupakan hasil factorial dri 4 yaitu 24.

1. Permutasi dan Kombinasi

* Codes :

#include<stdio.h>

//author : Evelyn Tjitrodjojo

//translator : M. Hilmi Ramadhan

int faktorial(int x);

int kombinasi(int x,int y);

int permutasi(int x, int y);

int main(){

int a,b;

scanf("%d %d",&a,&b);

printf("kombinasi = %d\n",kombinasi(a,b));

printf("permutasi = %d\n",permutasi(a,b));

}

int faktorial(int x)

{

int i,faktorial=1;

for(i=1;i<=x;i++)

{

faktorial=faktorial\*i;

}

return faktorial;

}

int kombinasi(int x, int y){

int komb;

komb=faktorial(x)/(faktorial(x-y)\*faktorial(y));

return komb;

}

int permutasi(int x, int y){

int permut;

permut=faktorial(x)/faktorial(x-y);

return permut;

}

* Penjelasan :

1. Pada dasarnya program ini seperti kalkulator khususnya menghitung factorial, kombinasi, dan permutasi.
2. Menggunakan for loop dengan sesuai batas yg dimasukkan oleh user.
3. Terdapat 3 fungsi yaitu factorial, kombinasi, dan permutasi.
4. Untuk factorial dilakukan perhitungan spt biasa contoh 5! = 5.4.3.2.1; sedangkan kombinasi menggunakan rumus n C k = 5C5 = 5! / (5-5)!.5! = 1.
5. Selanjutnya untuk permutasi dengan rumus n P k = 5 P 5 = 5!/(5-5)! = 120.

* Input : Angka test case yang akan dihitung baik itu dari kombinasi dan permutasi. Misal inputan 5 5
* Output: hasil dari kombinasi dan permutasi. Maka outputnya adalah kombinasi = 1 dan permutasi = 120

1. Faktorial Reskursif

* Codes :

#include<stdio.h>

//author : Evelyn Tjitrodjojo

int faktor(int);

int main()

{

int n;

scanf("%d",&n); //input angka faktorial printf("%d\n",faktor(n)); //print hasil fungsi faktorial

}

int faktor(int a)

{ //fungsi rekursif faktorial

if(a==1) return 1; //jika angka=1, langsung print 1 if(a%2==0)

if(a % 2 == 0)

{

a=(a/2)\*faktor(a-1); //bila angka genap maka dikali fungsi faktor

}

else

{a\*=faktor(a-1);} //jika angka dikali angka sebelumnya

return a;

}

* Penjelasan :
* Input :
* Output:

1. Transpose Matrix

* Codes :

#include<stdio.h>

//author : Evelyn Tjitrodjojo

//translator: M. Hilmi Ramadhan

int main ()

{

int i, j, t; //deklarasi variable yg digunakan

scanf("%d", &t); //input ordo

int mabok[t][t]; //deklarasi array 2d yang akan mjd matriks

for (i=0;i<t;i++) // perulangan utk baris matriks asli

{

for(j=0;j<t;j++)// perulangan utk kolom matriks asli

{

scanf("%d",&mabok[i][j]); //input isi matriks

}

getchar(); //mengambil enter

}

for (i=0;i<t;i++)// perulangan untuk baris matriks transpose

{

for(j=0;j<t;j++)// perulangan utk kolom matriks tranpose

{

if(j<t-1)

{

printf("%d ",mabok[j][i]); // proses penukaran

//print matriks transpose dengan spasi

}

else

printf("%d",mabok[j][i]); //print matriks transpose

}

printf("\n"); //print enter

}

}

* Penjelasan : (*https://www.youtube.com/watch?v=f\_v388iq048*)

1. Program ini bertujuan untuk melakukan invers pada matriks inputan user, dengan catatan ordo yang sama. Missal 3\*3.
2. Kemudian melakukan perulangan for loop pertama matriks untuk baris dan kolom dari matriks asli inputan user.
3. Selanjutnya distore ke mabok[i][j] (array 2d) dan masuk kembali ke perulangan untuk melakukan transpose (pembalikan dari baris mjd kolom dengan searah clockwise).
4. Setelah selesai maka ditampilkan ke layar.

* Input : Meminta inputan user untuk ordo matriks dan setiap per baris pada matriks yang akan di transpose (pertukaran antara baris dan kolom searah clockwise) nantinya.
* Output: Hasil dari transposean matriks yaitu pertukaran baris dan kolom.

1. Pembalik Angka

* Codes :

#include<stdio.h>

//author : Evelyn Tjitrodjojo

//translator by : M. Hilmi Ramdhan

int main()

{

int T,N,a,b;

printf("Masukan berapa banyak kali mencoba membalik bilangan\n");

scanf("%d",&T); //input test case berapa kali percobaan

for(a=0;a<T;a++)

{

scanf("%d",&N); //input angka yang akan jadi percobaanya(bias berupa urutan)

if(N==0)

{

printf("%d",N); //print angka 0 jika n=0

}

else

while (N>0) //Perulangan akan berlangsung hingga n = 0

{

b=N%10; //di mod 10 untuk mengetahui angka belakang

N=N/10; //angka dibalik 10

printf("%d",b); //print pembalik angka

}

printf("\n");

}

return 0;

}

* Penjelasan : (*https://www.youtube.com/watch?v=yPNV-VU7\_Y8*)

1. Program ini dibuat untuk melakukan pembalikan urutan angka dari inputan user yang sesuai juga dengan keinginan user berapa banyak melakukan pembalikan.
2. Prinsipnya adalah semua urutan angka inputan user di modulo atau sisa bagi dengan 10 terlebih dahulu untuk mengetahui angka yang paling belakang.
3. Setelah mengetahui maka angka tsb satu persatu di bagi oleh 10 dimulai dri paling belakang.
4. Setelah terus berulang selama while loop N > 0, sampai pada 0>0 maka loop selesai dan dapat di tampilkan ke layar.

* Input : Test case uji, seberapa banyak ingin melakukan pembalikan angka atau berapa kali program diinginkan untuk berjalan.
* Output: Berupa urutan angka yang berkebalikan dari inputan user.