who to convert for loop which is entry control loop into exist control

=>#include<stdio.h>

int main()

{

int itr;

itr=10;

for(itr=0;itr<10;itr++)

{

printf("\nHello");

}

return 0;

}

=>do

{

printf("\nHello");

itr++;

}while(itr<10);

printf("\n\n");

return 0;

}

\*/

=>#include<stdio.h>

int main()

{

for(int i=0;;)

{

printf("%d\n",i);

i++;

if(i>5)

{

break;

}

}

return 0;

}

=>goto:

#include<stdio.h>

int main()

{

int itr=1;

LoopLabel:

printf("\nitr=%d",itr);

itr++;

if(itr>5)

goto ExitLabel;

else

goto LoopLabel;

ExitLabel:

printf("\n\n");

return 0;

}

Arrays:\*we should use array with particular task only

\*[] is used to define arrays

\*arrays are homogeneous

\*when we are declaring the array we should be declare in consecutive memory.

\*EX: int arr[10] are these 10 elements are stored in consecutive memmory location for particular task only.

\*ArrName[indexValue]=value;

\*indexvalue is always whole number which is integer

=>struct Emp

{

int id;

char name[20];

};

struct Emp e[10];

\*here array is used only for store Emp structure that is id

=>how an array works internally

Baseaddress+(indexvalue\*sizeof(dt))

There are four types of arrays:

1.static:the size of array is known before to the compl time

EX: int arr[5];

2.dynamic :the size of the array is known at run time

EX: we malloc,calloc,realloc => .stdlib.h

3.stretchable:\*renaming the dynamic array in different way

\*it can be increased or decreased depending on the need for dynamic

Ex: malloc, calloc. realloc

4.mutable:\*size of the array is known at the time of linking and before execution

\*starting it is static at link time

=>to define two dimensional arrays

dt arrName[Row][Col];

int a[2][3]={{1,2,3}{4,5,6}};

int a1[2][3]={1,2,3,4,5,6};

1 2 3

4 5 6

2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011

1 2 3 4 5 6

a[0][0] a[0][1] a[0][2] a[1][0] a[1][1] a[1][2]

EX:

\*it is static way

#include <stdio.h>

#define ROW 2

#define COL 3

int main() {

int a[ROW][COL]={{1,2,3},{4,5,6}};

printf("\n%d",a[1][2]);

printf("\n\n");

return 0;

}

output:6

EX2:

#include <stdio.h>

#define ROW 2

#define COL 3

int main() {

int a[ROW][COL]={1,2,3,4,5,6};

printf("\n%d",a[1][2]);

printf("\n\n");

return 0;

}

=>#include <stdio.h>

#define ROW 2

#define COL 3

int main() {

int a1[ROW][COL]={{1,2,3},{4,5,6}};

int a2[ROW][COL]={1,2,3,4,5,6};

int riv,civ;

printf("\n%d\n");

for(riv=0;riv<ROW;riv++)

{

for(civ=0;civ<COL;civ++)

printf("%d",a2[ROW][COL]);

printf("\n");

}

printf("\n\n");

return 0;

}

=>#include <stdio.h>

#define CAP 100

int main()

{

int a[CAP], i;

int mid, countodd, counteven;

int m=51, n=103;

mid = (n-m)/2;

printf("\nmid=%d",mid);

for(i=m,countodd=0,counteven=mid;i<n;i++)

{

if(i%2!=0)

{

//odd

a[countodd]=i;

countodd++;

}

else

{

//even

a[counteven]=i;

printf("\na[%d]=%d",counteven,a[counteven]);

counteven++;

}

}

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

printf("%d ",a[i]);

}

\*to reverse an array

#include<stdio.h>

#define CAP 10

int main()

{

int i,t,mid;

int a[CAP]={1,2,3,4,5,6,7,8,9,10};

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

printf(" %d ",a[i]);

printf("\%d\n",a[a[2]]);

printf("\n%d\n",2[a]);

//logic rev array

for(i=0;mid=CAP/2,i<mid;i++)

{

t=a[i];

a[i]=a[(CAP-1)-i];

a[(CAP-1)-i]=t;

}

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

printf(" %d ",a[i]);

}

Functions:\*In functions we should always return and in rear case we use void

There are two types of function

1.std lib function:printf,sqrt,abs,pow

2.user defined:user is defining his/her own task to be performed

syntax: rdt fName(input args)

{

sts;

return rdt;

}

EX: int add(int a,int b)

{

int result=a+b;

return result;

}

prototype of these program

int add(int,int);

\*it should be in header file

and function definition in src file

// Online C compiler to run C program online

#ifndef ISPRIME\_H

#define ISPRIME\_H

#define TRUE 1

#define FALSE 0

int isprime(int);

#endif

//it will be in .h file which is header file

#include<isprime.h>

int isprime(int val)

{

int it,falg=0;

if(val<=1)

return False

for(it=2;it<=val/2;it++)

{

if(val%it==0)

{

flag=1;

break;

}

}

if(flag==1)

return False;

else

return True;

}

// it will be in src file

#include<stdio.h>

#include<isprime.h>

int main()

{

int num=17;

if(isprime(num)==1)

printf("\n%d a prime number:",num);

else

printf("\n%d is not a prime number:",num);

printf("\n\n")

return 0;

}

=>#include <stdio.h>

#include<stdlib.h>

int changevalue(int [],int);

void disp(int [],int);

int main() {

int a[5]={1,2,3,4,5};

printf("\nBA of arr(main): %u\n",&a[0]);

disp(a,5);

return 0;

}

void disp(int arr[],int n)

{

int i;

printf("\nBA of arr(disp):%u\n",&arr[0]);

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

printf("%d ",arr[i]);

printf("\n\n");

}

output:BA of arr(main): 2919680464

BA of arr(disp):2919680464

1 2 3 4 5

=>#include <stdio.h>

#include<stdlib.h>

int changevalue(int \*,int);

void disp(int [],int);

int main() {

int a[5]={1,2,3,4,5};

printf("\nBA of arr(main): %u\n",&a[0]);

disp(a,5);

return 0;

}

void disp(int \*arr,int n)

{

int i;

printf("\nBA of arr(disp):%u\n",&arr[0]);

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

printf("%d ",arr[i]);

printf("\n\n");

}

=>#include <stdio.h>

#include<stdlib.h>

int changevalue(int \*,int);

void disp(int [],int);

int main() {

int a[]={1,2,3,4,5};

int CAP=sizeof(a)/sizeof(a[0]);

printf("\nBA of arr(main): %u\n",&a[0]);

printf("\nCAP=%d\n",CAP);

disp(a,5);

return 0;

}

void disp(int \*arr,int n)

{

int i;

printf("\nBA of arr(disp):%u\n",&arr[0]);

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

printf("%d ",arr[i]);

printf("\n\n");

}

=>#include <stdio.h>

#include<stdlib.h>

int changevalue(int [],int);

void disp(int [],int);

int main() {

int a[]={1,2,3,4,5};

int CAP=sizeof(a)/sizeof(a[0]);

printf("\nBA of arr(main): %u\n",&a[0]);

printf("\nCAP=%d\n",CAP);

disp(a,5);

changevalue(a,3);

disp(a,5);

return 0;

}

void disp(int arr[],int n)

{

int i;

printf("\nBA of arr(disp):%u\n",&arr[0]);

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

printf("%d ",arr[i]);

printf("\n\n");

}

int changevalue(int \*arr,int key)

{

int i;

int flag=0;

printf("\nCAP=%d",sizeof(arr)/sizeof(arr[0]));

for(i=0;i<5;i++)

{

if(arr[i]==key)

{

arr[i]=40;

flag=1;

}

}

if(flag==0)

return 1;

else

return 0;

}