**C program using pointers and arrays**

//C program using pointers and arrays

#include<stdio.h>

int main(){

int n,i,opt,\*ptr,sum=0,sq;

printf("The square of the given numbers will be added\n");

printf("Enter number of elements:");

scanf("%d",&n);

int arr[n];

printf("Enter your elements:");

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

{

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

}

ptr=arr;

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

{

for(int j=0;j<2;j++)

{

sq=(\*ptr)\*(\*ptr);

}

sum=sum+sq;

ptr++;

}

printf("The sum of square of given numbers is %d",sum);

return 0;

}

**C program pass arrays to function**

//C program pass arrays to function

#include<stdio.h>

#include<stdlib.h>

int b[100][100],k=0;

void print(int a[],int n)

{

int i;

printf("The array is now: ");

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

{

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

}

printf("\nThe number of elements in the array is %d\n",n);

}

int sort(int a[],int n)

{

int opt,i,j,t;

while(1){

printf("1.Ascending order\n2.Desending order\n");

scanf("%d",&opt);

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

{

b[k][j]=a[j];

}

switch (opt)

{

case 1:for(i=0;i<n;i++)

{

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

{

if(a[i]>a[j])

{

t=a[i];

a[i]=a[j];

a[j]=t;

}

}

}

return 0;

case 2: for(i=0;i<n;i++)

{

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

{

if(a[i]<a[j])

{

t=a[i];

a[i]=a[j];

a[j]=t;

}

}

}

return 0;

default:printf("invalid option!!\n");

break;

}

}

}

void undo(int a[],int n)

{

if(k<0)

{

printf("\nNo previous history\n");

}

else{

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

{

a[j]=b[k][j];

}

}

}

int main()

{

int n,i,opt;

printf("This program will undo the changes done to an array\n");

printf("Enter number of elements in your array:");

scanf("%d",&n);

int arr[n];

printf("Enter your elements:");

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

{

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

}

while(1){

printf("\n1.Sort the array\n2.Undo\n3.Exit\nEnter option:");

scanf("%d",&opt);

switch (opt)

{

case 1:sort(arr,n);

print(arr,n);

k++;

break;

case 2:k--;

undo(arr,n);

print(arr,n);

break;

case 3:exit(0);

default:printf("invalid option!!\n");

break;

}

}

return 0;

}

**C program to pass structure to a function**

//C program to pass structure to a function

#include<stdio.h>

void print(int ans)

{

printf("%d",ans);

}

void add(int a, int b)

{

int ans;

ans=a+b;

print(ans);

}

void sub(int a, int b)

{

int ans;

ans=a-b;

print(ans);

}

void mul(int a, int b)

{

int ans;

ans=a\*b;

print(ans);

}

void div(int a, int b)

{

int ans;

ans=a/b;

print(ans);

}

int main()

{

int opt;

struct calc

{

int a;

int b;

}s;

while(1){

printf("\nEnter number 1:\n");

scanf("%d",&s.a);

printf("Enter number 2:\n");

scanf("%d",&s.b);

printf("Select the operation to be done\n");

printf("1.add\n2.sub\n3.multiply\n4.divide\n5.exit\n");

scanf("%d",&opt);

switch (opt)

{

case 1:add(s.a,s.b);

break;

case 2:;sub(s.a,s.b);

break;

case 3:mul(s.a,s.b);

break;

case 4:div(s.a,s.b);

break;

case 5:return 0;

break;

default:printf("Invalid option!!");

break;

}

}

}

**C program with pointers with structures**

//C program with pointers with structures

#include<stdio.h>

int main ()

{

struct BMI

{

float h;

char name[20];

float w;

float bmi;

}s1,\*s;

printf("Enter name:\n");

scanf("%s",&s1.name);

printf("Enter weight in kgs:\n");

scanf("%f",&s1.w);

printf("Enter height in meters:\n");

scanf("%f",&s1.h);

s=&s1;

s->bmi=s->w/(s->h\*s->h);

if(s->bmi<18.5)

{

printf("%s is uderweight",s->name);

}

else if(s->bmi<=24.9)

{

printf("%s is fit",s->name);

}

else if(s->bmi<=29.9)

{

printf("%s is overweight",s->name);

}

else if(s->bmi<=34.9)

{

printf("%s is uder class 1 obesity",s->name);

}

else if(s->bmi<=39.9)

{

printf("%s is uder class 2 obesity",s->name);

}

else

{

printf("%s is uder class 3 obesity",s->name);

}

return 0;

}

**C program with self referential structures**

//C program with self referential structures

#include<stdio.h>

int main()

{

int i,n,m,flag=0;

printf("This program will show who sits before and after you\n");

printf("Enter number seats :");

scanf("%d",&n);

struct seat

{

int num;

char name[20];

struct seat\* prev\_link;

struct seat\* next\_link;

}s[n];

s[0].prev\_link = NULL;

s[n-1].next\_link = NULL;

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

{

printf("Enter number %d : ",i+1);

scanf("%d",&s[i].num);

printf("Enter name %d : ",i+1);

scanf("%s",&s[i].name);

if(i>0)

{

s[i].prev\_link=&s[i-1];

}

if(i<n-1)

{

s[i].next\_link=&s[i+1];

}

}

printf("To get information about your seat\n") ;

printf("Enter your seat number:");

scanf("%d",&m);

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

{

if(s[i].num==m)

{

flag=1;

break;

}

}

if(flag==1)

{

if(i==0)

{

printf("Your seat is before %s's seat",s[i].next\_link->name);

}

else if(i==n-1)

{

printf("Your seat is behind %s's seat",s[i].prev\_link->name);

}

else

{

printf("Your seat is behind %s's seat and before %s's seat",s[i].prev\_link->name,s[i].next\_link->name);

}

}

else

{

printf("There is no such seat in the given number");

}

return 0;

}

**C program with call by reference on structures**

//C program with call by reference on structures.

#include<stdio.h>

struct student

{

long roll;

char name[20];

int mark;

}s[100];

void new(int n)

{

printf("Enter roll.no:");

scanf("%ld",&s[n].roll);

printf("Enter name:");

scanf("%s",&s[n].name);

printf("Enter mark:");

scanf("%d",&s[n].mark);

}

void print(int n)

{

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

{

printf("%ld %s %d\n",s[i].roll,s[i].name,s[i].mark);

}

}

void search(int n)

{

long num;

int pos=0;

printf("Enter the roll no of the student:");

scanf("%ld",&num);

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

{

if(s[i].roll==num)

{

pos=i+1;

break;

}

}

if(pos)

{

printf("Found entry\nRoll no: %ld\nName : %s\nMarks secured : %d",s[pos-1].roll,s[pos-1].name,s[pos-1].mark);

}

else

{

printf("No entry found!!!");

}

}

int main()

{

int n=0,opt;

printf("In this program you can enter and search the details of the entered students");

while(1){

printf("\n1.New entry\n2.Print entry\n3.search details\n4.exit\n");

scanf("%d",&opt);

switch (opt)

{

case 1: new(n);

n++;

break;

case 2: print(n);

break;

case 3: search(n);

break;

case 4: return 0;

default:printf("invalid option!!\n");

break;

}

}

return 0;

}

**C program for complete Array ADT operations**

//C program for complete Array ADT operations

#include<stdio.h>

#include<stdlib.h>

int n;

void print(int a[])

{

int i;

printf("The array is now: ");

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

{

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

}

printf("\nThe number of elements in the array is %d\n",n);

}

int insert(int arr[])

{

int opt,num,pos;

while(1){

printf("1.At the begining\n2.In the middle\n3.At the end\n4.Return\n");

scanf("%d",&opt);

switch (opt)

{

case 1:for(int i=n;i>=0;i--)

{

arr[i]=arr[i-1];

}

printf("Enter the element to be insert:");

scanf("%d",&num);

arr[0]=num;

n++;

break;

case 2: printf("At what position the element should be inserted:");

scanf("%d",&pos);

printf("Enter the element should be inserted:");

scanf("%d",&num);

for(int i=n;i>=pos;i--)

{

arr[i]=arr[i-1];

}

arr[pos-1]=num;

n++;

break;

case 3: printf("Enter the element to be inserted:");

scanf("%d",&num);

arr[n]=num;

n++;

break;

case 4: return 0;

default:printf("invalid option!!\n");

break;

}

if(opt<4)

{print(arr);}

}

}

void search(int arr[])

{

int num,flag=0,i;

printf("Enter the element to be searched:");

scanf("%d",&num);

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

{

if(arr[i]==num)

{

printf("The element is found to be at %d position of the array\n",i+1);

flag=1;

}

}

if(flag==0){

printf("The element is not found in the array\n");

}

}

int delete(int arr[])

{

int opt,pos;

while(1){

printf("1.At the begining\n2.In the middle\n3.At the end\n4.Return\n");

scanf("%d",&opt);

switch (opt)

{

case 1:for(int i=1;i<n;i++)

{

arr[i-1]=arr[i];

}

n--;

break;

case 2: printf("At what position the element should be deleted:");

scanf("%d",&pos);

if(pos>n){printf("Invalid position!");}

else{

for(int i=pos;i<n;i++)

{

arr[i-1]=arr[i];

}

n--;}

break;

case 3: n--;

break;

case 4: return 0;

default:printf("invalid option!!\n");

break;

}

if(opt<4)

{print(arr);}

}

}

int sort(int a[])

{

int opt,i,j,t;

while(1){

printf("1.Ascending order\n2.Desending order\n");

scanf("%d",&opt);

switch (opt)

{

case 1:for(i=0;i<n;i++)

{

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

{

if(a[i]>a[j])

{

t=a[i];

a[i]=a[j];

a[j]=t;

}

}

}

return 0;

case 2: for(i=0;i<n;i++)

{

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

{

if(a[i]<a[j])

{

t=a[i];

a[i]=a[j];

a[j]=t;

}

}

}

return 0;

default:printf("invalid option!!\n");

break;

}

}

}

int main()

{

int i,opt;

printf("Enter number of elements in your array:");

scanf("%d",&n);

int arr[n];

printf("Enter your elements:");

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

{

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

}

while(1)

{

printf("1.insert an element\n2.Delete an element\n3.Search an element\n4.Sort the array\n5.Print\n6.Create New array\n7.Exit\n");

scanf("%d",&opt);

switch (opt)

{

case 1:insert(arr);

break;

case 2:delete(arr);

break;

case 3:search(arr);

break;

case 4:sort(arr);

print(arr);

break;

case 5:print(arr);

break;

case 6: printf("Enter number of elements in your array:");

scanf("%d",&n);

printf("Enter your elements:");

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

{

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

}

break;

case 7:exit(0);

default:printf("invalid option!!\n");

break;

}

}

return 0;

}