Task-1

#include<iostream>

using namespace std;

struct person {

int age, weight;

};

int main() {

person per;

int a, b;

int \*ptrA, \*ptrB;

cout << "Enter a = ";

cin >> a;

cout << "Enter b = ";

cin >> b;

ptrA = &a;

ptrB = &b;

per.age = \*ptrA;

per.weight = \*ptrB;

cout << "PtrA = " << \*ptrA << endl;

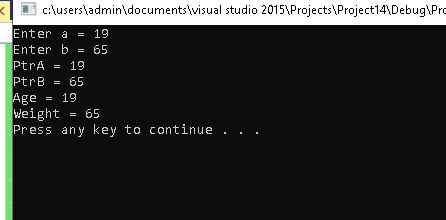
cout << "PtrB = " << \*ptrB << endl;

cout << "Age = " << per.age << endl;

cout << "Weight = " << per.weight << endl;

system("pause");

}



Task-2

#include<iostream>

using namespace std;

struct measurement {

int feet;

float inch;

};

void AddLength(measurement &a, measurement &b) {

measurement d3;

d3.feet = a.feet + b.feet;

d3.inch = a.inch + b.inch;

cout << d3.feet <<" feet and "<< d3.inch << " inches" << endl;

}

int main() {

measurement d1;

measurement d2;

measurement \*d3;

cout << "Enter D1 Atrributes" << endl;

cout << "Enter Feet: ";

cin >> d1.feet;

cout << "Enter inch: ";

cin >> d1.inch;

cout << "Enter D2 Atrributes" << endl;

cout << "Enter Feet: ";

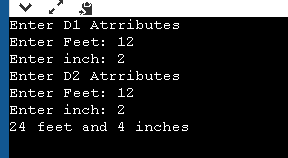
cin >> d2.feet;

cout << "Enter inch: ";

cin >> d2.inch;

AddLength(d1, d2);

}



Task-03

#include<iostream>

using namespace std;

struct Address{

string street,city,state;

};

struct Employee {

int employeenumber;

string name,destination;

Address \*address;

};

int main(){

Employee e[5];

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

cout<<"Enter Emp No: ";

cin>>e[i].employeenumber;

cout<<"Enter Emp Name: " ;

cin>>e[i].name;

cout<<"Enter Emp destination: ";

cin>>e[i].destination;

cout<<"Enter Emp street: ";

cin>>e[i].address->street;

cout<<"Enter Emp city: ";

cin>>e[i].address->city;

cout<<"Enter Emp State: ";

cin>>e[i].address->state;

}

int search;

cout<<"Search Employee no: ";

cin>>search;

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

if(search == e[i].employeenumber){

cout<<"Emp No:"<<e[i].employeenumber<<endl;

cout<<"Emp Name:"<<e[i].employeenumber<<endl;

cout<<"Emp Destination:"<<e[i].employeenumber<<endl;

cout<<"Emp street:"<<e[i].address->street<<endl;

cout<<"Emp city:"<<e[i].address->city<<endl;2

cout<<"Emp state:"<<e[i].address->state<<endl;

}

}

system("pause");

}

Output



Task-4

#include<iostream>

using namespace std;

struct Items{

int code;

string description;

int stock;

};

void ItemInput(Items \*item,int n){

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

cout<<"Enter code: ";

cin>>item[i].code;

cout<<"Enter description: ";

cin>>item[i].description;

cout<<"Enter Stock: ";

cin>>item[i].stock;

}

}

int main(){

Items item[10];

ItemInput(item,3);

for (int i=0;i<3;i++){

cout<<"Code:";

cout<<item[i].code<<endl;

cout<<"Description: ";

cout<<item[i].description;

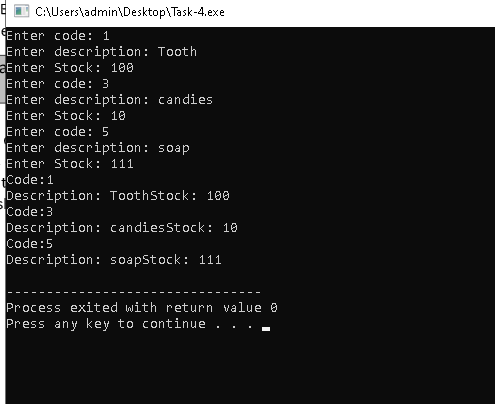
cout<<"Stock: ";

cout<<item[i].stock;

cout<<endl;

}

}



Task-5

#include<iostream>

using namespace std;

class Array{

int \*arr;

int s;

static int n;

public:

Array(int size = 5){

s = size;

arr = new int[size];

for (int i=0;i<s;i++){

arr[i] = 0;

}

}

void printList(){

for (int i=0;i<s;i++){

cout<<arr[i]<<" ";

}

cout<<endl;

}

void insertElement(int elem){

arr[n] = elem;

n++;

}

void insertElementAtPos(int elem, int pos){

arr[pos] = elem;

}

int searchElement(int elem){

int search;

for (int i=0;i<s;i++){

if(elem == arr[i]){

search = i;

return search;

}

}

}

bool deleteElement(int elem){

int index = 0;

index = searchElement(elem);

if(index!=0){

arr[index] = 0;

return true;

}

else

return false;

}

bool isFull(){

if(n == s-1)

return true;

return false;

}

int Length(){

return s;

}

bool IsEmpty(){

if(n ==0)

return true;

return false;

}

void reverseList(){

int \*temp = new int[n];

for (int i=0.j=n;i<s;i++,j--){

swap(arr[i],arr[j]);

}

}

void EmptyList(){

for (int i=0;i<s;i++){

arr[i] = 0;

}

}

void copyList(int \*copy){

for (int i=0;i<s;i++){

copy[i] =arr[i] ;

}

}

void sort(){

int temp = 0;

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

{

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

{

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

{

temp =arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

bool compare(Array \*arr1,,Array \*arr2){

bool check = false;

if(arr1->Length() == arr2->Length()){

for (int i=0;i<size;i++){

if(arr[i]==arr2[i])

check = true;

else{

check = false;

return check

}

}

}

else

return false;

}

};

int Array :: n = 0;

int main(){

}