**Dynamic array**

#include <iostream>

using namespace std;

class DArray{

private:

//array

int\* arrayNum=NULL;

int size; //array size (no. of elements in an array)

int capacity; //container size

public:

//constructor

DArray(){

capacity=1; //default capacity is 1 only one container

size=0;

//allocate memory to the array

arrayNum=new int[size];

}

//constructor - taking size from the user

DArray(int capacity){

this->capacity=capacity;

arrayNum=new int[capacity];

size=0; //no elements in an array

}

//growArray method

void growArray(){

//update capacity

capacity=capacity\*2; //double the size

//temp array

int\* temp=new int[capacity];

//copy data in new array

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

temp[i]=arrayNum[i];

}

//deallocate the memory of previous array

delete[] arrayNum;

arrayNum=temp;

}

//shrinkArray method

void shrinkArray(){

//update array capacity - decrease

capacity = size;

//temp new array

int\* temp=new int[capacity];

//copy data from original array to the new temp array

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

temp[i]=arrayNum[i];

}

//deallocate memory

//delete old array

delete[] arrayNum;

arrayNum=temp;

}

//push\_back method - insert an element

void push\_back(int data){

//check if array has capacity to save new data

//update capacity

if(size==capacity){

growArray();//update capacity

}

//insert data at the loc 'size'

arrayNum[size]=data;

//increment size

size++;

}

//pop\_back method - delet an element

//deleting element at last index

void pop\_back(){

if(isEmpty){

cout<<”Array is empty!!”<<endl;

}

esle{

arrayNum[size-1]=0;

size--;

//update capacity

if(size==(capacity/2)){

shrinkArray();}

}

}

}

//insert element at given index

void insertAt(int index, int value){

if(size==capacity)

growArray();

for(int i=size-1; i>=index; i--)

arrayNum[i+1]=arrayNum[i];

arrayNum[index]=value;

size++;

}

//delete element at given index

void deleteAt(int index){

for(int i=index; i<size; i++)

arrayNum[i]=arrayNum[i+1];

arrayNum[size-1]=0;

size--;

//reduce size of array

if(size==(capacity/2))

shrinkArray();

}

//isEmpty

bool isEmpty(){

return size==0;

}

//search() - searching element in a given array

//return index number of the element found

int search(int value){

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

if(arrayNum[i]==value)

return i;

}

return -1;

}

//printArray details

void printDetails(){

//array capcaity

cout<<"Capacity of the is : "<<capacity<<endl;

//array total no. of elements

cout<<"Total number of elements in an array are : "<<size<<endl;

//display elements of array

cout<<"Array elements are : "<<endl;

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

cout<<arrayNum[i]<<" ";

}

cout<<endl;

}

};

int main() {

//creating obj

DArray da1;

// da1.printDetails();

da1.push\_back(1);

da1.push\_back(2);

da1.push\_back(3);

da1.printDetails();

// da1.pop\_back();

// da1.printDetails();

da1.insertAt(1,100);

da1.insertAt(0,9);

cout<<"Array\' data after inserting new values : "<<endl;

da1.printDetails();

//delete

da1.deleteAt(3);

cout<<"Array\'s data after deleting a value at index 3 : "<<endl;

da1.printDetails();

return 0;

}