**404. Data Structure Programs**

## Name – Girase Prajakta Hitendra

## Class :- SYBCA Roll no. 24

**//function for Array Traversal:-**

**#include <iostream>**

**using namespace std;**

**void display(int arr[] , int n)**

**{**

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

**{**

**cout<<arr[i]<<"\t";**

**}**

**cout<<"\n";**

**}**

**// function for Array Insertion**

**int Insertion(int arr[] , int n , int capacity , int element , int ind)**

**{**

**if(n >= capacity)**

**{**

**return -1;**

**}**

**for(int i = n; i >= ind; i--)**

**{**

**arr[i+1] = arr[i];**

**}**

**arr[ind] = element;**

**cout<<"\n";**

**return 0;**

**}**

**int main() {**

**int arr[10] = {1 , 2 , 3 , 4 , 5};**

**int n = 5;**

**display(arr , n);**

**Insertion(arr, n , 10 , 58 , 1);**

**display(arr , n+1);**

**return 0;**

**}**

**// Traversing : Visit each and every element of an array.**

**// program : - >**

**// #include <iostream>**

**// using namespace std;**

**// int main() {**

**// int len;**

**// cout<<"Enter length of an Array : ";**

**// cin>>len;**

**// int arr[len];**

**// for(int i = 0; i < len; i++){**

**// cout<<"Enter "<<i+1<<" Element : ";**

**// cin>>arr[i];**

**// }**

**// for(int i = 0; i < len; i++) {**

**// cout<<i+1<<" : "<<arr[i]<<endl;**

**// }**

**// return 0;**

**// }**

1. **Matrix Program:-**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int i,j,row,col,a[100][100],b[100][100],sum[100][100];**

**cout << "Enter the Row" <<endl;**

**cin >> row;**

**cout<<"Enter the Columns"<<endl;**

**cin>>col;**

**// sotring of 1st matrix entered by user**

**cout<<"Enter the 1st matrix"<<endl;**

**for(i=0;i<row; i++){**

**for(j=0; j<col;j++){**

**cin>>a[i][j];**

**}**

**}**

**cout<<"the 1st matrix"<<endl;**

**for(i=0;i<row; i++){**

**for(j=0; j<col;j++){**

**cout<<a[i][j]<<" "<<endl;**

**}**

**cout<<" ";**

**}**

**// sorting of 2nd matrix entered by user**

**cout<<"Enter the 2nd matrix"<<endl;**

**for(i=0; i< row; i++){**

**for(j=0; j < col; j++){**

**cin>>b[i][j];**

**}**

**}**

**cout<<"the 2nd matrix"<<endl;**

**for(i=0;i<row;i++){**

**for(j=0;j<col; j++){**

**cout<<b[i][j]<<" "<<endl;**

**}**

**}**

**//adding matrix**

**for(i=0; i< row; ++i){**

**for(j=0; j< col; ++j){**

**sum[i][j]=a[i][j] + b[i][j];**

**}**

**}**

**//diplay the sum of matrix**

**cout<<"The sum of two matrix is : "<<endl;**

**for(i=0; i<row; i++){**

**for(j=0;j<col; j++){**

**cout<<sum[i][j]<<" ";**

**if(j==col-1){**

**cout<<endl;**

**}**

**}**

**}**

**return 0;**

**}**

**//Searching of Array:-**

**2)Linear Search Program :-**

**#include<iostream.h>**

**#include<conio.h>**

**void main(){**

**int arr[5],i,num,index;**

**cout<<"Enter the number"<<endl;**

**for(i=0;i<10;i++){**

**cin>>arr[i];**

**}**

**cout<<"Enter the to search : "<<endl;**

**cin>>num;**

**for(i=0;i<10;i++){**

**if(arr[i]==num){**

**index=i;**

**break;**

**}**

**}**

**cout<<num<<" is Found at Index No : "<<index;**

**cout<<endl;**

**getch();**

**clrscr();**

**}**

**3) Binary Search Program :-**

**#include <iostream>**

**using namespace std;**

**int binarySearch(int arr2[] , int n , int d) {**

**int i = 0;**

**while(i <= n) {**

**int mid = (i + n) / 2;**

**if(arr2[mid] == d) {**

**return mid;**

**}**

**else if(d > arr2[mid]) {**

**i = mid + 1;**

**}**

**else {**

**n = mid - 1;**

**}**

**}**

**return -1;**

**}**

**int main() {**

**int len;**

**int dele;**

**cout<<"Enter length of an Array : ";**

**cin>>len;**

**int arr[len];**

**for (int i = 0; i < len; i++) {**

**cout<<"Enter "<<i+1<<" Element : ";**

**cin>>arr[i];**

**}**

**for (int i = 0; i < len; i++) {**

**cout<<i+1<<" : "<<arr[i]<<endl;**

**}**

**for (int i = 0; i < len; i++) {**

**for (int j = i+1; j < len; j++) {**

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

**int temp = arr[i];**

**arr[i] = arr[j];**

**arr[j] = temp;**

**}**

**}**

**}**

**cout<<"Sorted"<<endl;**

**for (int i = 0; i < len; i++) {**

**cout<<i+1<<" : "<<arr[i]<<endl;**

**}**

**cout<<"Enter Element to find in Array : ";**

**cin>>dele;**

**int result = binarySearch(arr , len , dele);**

**if(result == -1) {**

**cout<<"Element Not Found !!!";**

**}**

**else {**

**cout<<dele<<" Found At Index : "<<result;**

**}**

**return 0;**

**}**

**// Linear Search and Binary Search in one program : ->**

**#include <iostream>**

**using namespace std;**

**// Function for Linear Searching**

**int linearSearch(int arr[] , int n , int element) {**

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

**if(arr[i] == element) {**

**return i;**

**}**

**}**

**return -1;**

**}**

**// Function for Binary Searching**

**int binarySearch(int arr[] , int n , int element ) {**

**int lb = 0;**

**int ub = n;**

**while(lb<=ub) {**

**int mid = ( lb + ub ) / 2;**

**if(arr[mid] == element) {**

**return mid;**

**}**

**if(arr[mid] < element) {**

**lb = mid + 1;**

**}**

**else {**

**ub = mid - 1;**

**}**

**}**

**return -1;**

**}**

**int main() {**

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

**int n = sizeof(arr)/sizeof(int);**

**int element = 2;**

**// int result = linearSearch(arr , n , element );**

**int result = binarySearch(arr , n , element);**

**if(result == -1) {**

**cout<<"Element not found !!!";**

**}**

**else {**

**cout<<element<<" Found at index : "<<result;**

**}**

**return 0;**

**}**

**//Sorting of Arrary:-**

**4)Bubble Sort Program:-**

**#include<iostream.h>**

**#include<conio.h>**

**void main()**

**{**

**int n,i,j,temp;**

**int a[50];**

**cout<<"Enter the Size of array"<<endl;**

**cin>>n;**

**cout<<"Enter the array Elements"<<endl;**

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

**cin>>a[i];**

**}**

**cout<<"Element before sort"<<endl;**

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

**cout<<a[i];**

**cout<<endl;**

**}**

**for(i=0;i<n-1;i++){**

**for(j=0;j<n-1;j++){**

**if(a[j]>a[j+1]){**

**temp=a[j];**

**a[j]=a[j+1];**

**a[j+1]=temp;**

**}**

**}**

**}**

**cout<<"Element after sort"<<endl;**

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

**cout<<a[i]<<endl;**

**}**

**getch();**

**}**

**5) Optimized Bubble sort Program:-**

**#include<iostream.h>**

**#include<conio.h>**

**void main(){**

**int i,j,temp,n;**

**int flag=0;**

**int a[50];**

**cout<<"Enter the size of array"<<endl;**

**cin>>n;**

**cout<<"Enter the array elements"<<endl;**

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

**cin>>a[i];**

**}**

**cout<<"Before sort"<<endl;**

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

**cout<<a[i]<<endl;**

**}**

**for(i=0;i<n-1;i++){**

**for(j=0;j<n-1;j++){**

**if(a[j]>a[j+1]){**

**temp=a[j];**

**a[j]=a[j+1];**

**a[j+1]=temp;**

**flag=1;**

**}**

**}**

**if(flag==0){**

**break;**

**}**

**}**

**cout<<"After Sort"<<endl;**

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

**cout<<a[i]<<endl;**

**}**

**getch();**

**}**

**6) Insertion Sort Program :-**

**#include<iostream.h>**

**#include<conio.h>**

**void main()**

**{**

**int i,j,temp,n;**

**int a[50];**

**cout<<"Enter the size of Array"<<endl;**

**cin>>n;**

**cout<<"Enter the Elements"<<endl;**

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

**{**

**cin>>a[i];**

**}**

**cout<<"Elements Before sort : "<<endl;**

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

**cout<<a[i]<<endl;**

**}**

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

**{temp=a[i];**

**j=i-1;**

**while(j>=0 && a[j]>temp)**

**{**

**a[j+1]=a[j];**

**j--;**

**}**

**a[j+1]=temp;**

**}**

**cout<<"Elements After Sort"<<endl;**

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

**{**

**cout<<a[i]<<endl;**

**}**

**getch();**

**clrscr();**

**}**

**7) Selection Sort :-**

**#include<iostream.h>**

**#include<conio.h>**

**void main()**

**{**

**int i,j,n;**

**int a[50];**

**cout<<"Enter the size of array"<<endl;**

**cin>>n;**

**cout<<"Enter the array elements"<<endl;**

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

**cin>>a[i];**

**}**

**cout<<"Before Sort"<<endl;**

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

**cout<<a[i]<<endl;**

**}**

**for(i=0;i<n-1;i++){**

**int min=i;**

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

**{**

**if(a[j]<a[min])**

**{**

**min=j;**

**}**

**}**

**if(min!=i)**

**{**

**int temp=a[i];**

**a[i]=a[min];**

**a[min]=temp;**

**}**

**}**

**cout<<"Sorted Array is : " <<endl;**

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

**cout<<a[i]<<endl;**

**}**

**getch();**

**clrscr();**

**}**

**8) QuickSort Program :-**

**#include<iostream.h>**

**#include<conio.h>**

**void swap(int \*a,int \*b)**

**{**

**int t=\*a;**

**\*a=\*b;**

**\*b=t;**

**}**

**int partition(int arr[],int low,int high)**

**{**

**int pivot=arr[high];**

**int i=(low-1);**

**for(int j=low;j<=high-1;j++)**

**{**

**if(arr[j]<pivot)**

**{**

**i++;**

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

**}**

**}**

**swap(&arr[i+1],&arr[j]);**

**return(i+1);**

**}**

**void quicksort(int arr[],int low,int high)**

**{**

**if(low<high)**

**{**

**int pi=partition(arr,low,high);**

**quicksort(arr,low,pi-1);**

**quicksort(arr,pi+1,high);**

**}**

**}**

**void main()**

**{**

**int a[10],n,i,j;**

**cout<<"enter size of array :"<<endl;**

**cin>>n;**

**cout<<"Enter the array Elements :"<<endl;**

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

**{**

**cin>>a[i];**

**}**

**quicksort(a,0,n-1);**

**cout<<"Array after sorting"<<endl;**

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

**{**

**cout<<a[i]<<" "<<endl;**

**}**

**getch();**

**clrscr();**

**}**

**9) merge Sort Program : -**

**10) STACK PROGRAM :-**

**#include <iostream>**

**using namespace std;**

**int stack=[5];**

**int top=1;**

**void push(){**

**int x;**

**cout<<"Enter the value"<<endl;**

**cin>>x;**

**if(top==4-1){**

**cout<<"Overflow";**

**}**

**else**

**{**

**top++;**

**stack[top]=x;**

**}**

**}**

**void pop(){**

**int data;**

**if(top==-1){**

**cout<<"Underflow"<<endl;**

**}**

**else{**

**data=stack[top];**

**top--;**

**}**

**}**

**void peek(){**

**if(top==-1){**

**cout<<"Stack is empty"<<endl;**

**}**

**else{**

**cout<<stack[top]<<endl;**

**}**

**}**

**void display(){**

**for(int i=top;i>=0;i--){**

**cout<<stack[i]<<endl;**

**}**

**}**

**int main()**

**{**

**int ch;**

**do{**

**cout<<"Enter the choice 1:push 2:pop 3:peek 4:display"<<endl;**

**cin>>ch;**

**switch(ch){**

**case 1 :push();**

**break;**

**case 2 :pop();**

**break;**

**case 3 :peek();**

**break;**

**case 4 :display();**

**break;**

**default:**

**cout<<"Invalid number"<<endl;**

**}**

**}**

**while(ch!=0);**

**return 0;**

**}**

**11) Queue Program :-**

**#include<iostream.h>**

**#include<conio.h>**

**int queue[5],n=100,front=-1,rear=-1;**

**void insert()**

**{**

**int val;**

**if(rear==n -1)**

**{**

**cout<<"Queue is Overflow" <<endl;**

**}**

**else**

**{**

**if(front==-1)**

**front=0;**

**cout<<"Insert the element in queue"<<endl;**

**cin>>val;**

**rear++;**

**queue[rear]=val;**

**}**

**}**

**void Delete()**

**{**

**if(front==-1 || front>rear)**

**{**

**cout<<"Queue is Underflow"<<endl;**

**return;**

**}**

**else{**

**cout<<"Element deleted from queue is"<<queue[front]<<endl;**

**front++;**

**}**

**}**

**void Display()**

**{**

**if(front==-1){**

**cout<<"queue is empty"<<endl;**

**}**

**else**

**{**

**cout<<"Queue elements are :-"<<endl;**

**for(int i=front;i<=rear;i++){**

**cout<<queue[i]<<" ";**

**cout<<endl;**

**}**

**}**

**}**

**void main(){**

**int ch;**

**cout<<"1) Insert element in queue 2)delete elements from queue 3)display elements of queue 4) Exit "<<endl;**

**do{**

**cout<<"Enter your Choice "<<endl;**

**cin>>ch;**

**switch(ch)**

**{**

**case 1: insert();**

**break;**

**case 2: Delete();**

**break;**

**case 3: Display();**

**break;**

**case 4: cout<<"Exit"<<endl;**

**break;**

**default: cout<<"Invalid Choice"<<endl;**

**}**

**}while(ch!=4);**

**getch();**

**clrscr();**

**}**