Term work

of

**Data Structures Lab (PCS-302)**

Submitted in partial fulfillment of the requirement for the III semester

**Bachelor of Technology**

By

**Mohd Armaan**

**2261368**

**Under the Guidance of**

**Mr. Devesh Pandey**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**GRAPHIC ERA HILL UNIVERSITY, BHIMTAL CAMPUS**

**SATTAL ROAD, P.O. BHOWALI**

**DISTRICT- NAINITAL-263132**

**2022-2023**



**CERTIFICATE**

**The term work of Data Structures Lab (PCS-302), being submitted by Mohd Armaan s/o Mohd Juber, University Roll Number 2261368 to Graphic Era Hill University Bhimtal Campus for the award of bona fide work carried out by him/her. He/She has worked under my guidance and supervision and fulfilled the requirement for the submission of this work report.**

**(…………………) (……………………)**

**Subject Professor HOD, CSE Dept.**



**ACKNOWLEDGEMENT**

I take immense pleasure in thanking Honorable **Mr. Devesh Pandey** for allowing us to carry out this project work under his excellent and optimistic supervision. This has all been possible due to his novel inspiration, able guidance and useful suggestions that have helped me in developing my subject concepts as a student.

I want to extend thanks to our President **“Prof. (Dr.) Kamal Ghanshala”** for providing us all infrastructure and facilities to work in need without which this work would not be possible.

**(Mohd Armaan)**

mohammadarmaan533@gmail.com



**STUDENT’S DECLARATION**

I, Mohd Armaan, hereby declare the work, which is being presented in the report, entitled **Term work of Data Structures Lab (PCS-302)**  in partial fulfillment of the requirement for the award of the degree **Bachelor of Technology (Computer Science)**  in the session **2023-2024** for semester III, is an authentic record of my own work carried out under the supervision of **Mr. Devesh Pandey**   
(Graphic Era Hill University, Bhimtal)

The matter embodied in this project has not been submitted by me for the award of any other degree.

Date: ………… ……………….

(Full signature of student)



**Computer Science and Engineering Department**

**Data Structures Lab (PCS-302)**

**Requirements:**

* Unix/Linux based Computer System

**Index/List of Experiments**

|  |  |  |  |
| --- | --- | --- | --- |
| 1. **No.** | **Practical Name** | **Date of Execution** | **Faculty Sign.** |
| **1** |  |  |  |
| **2** |  |  |  |
| **3** |  |  |  |
| **4** |  |  |  |
| **5** |  |  |  |
| **6** |  |  |  |
| **7** |  |  |  |
| **8** |  |  |  |
| **9** |  |  |  |
| **10** |  |  |  |
| **11** |  |  |  |
| **12** |  |  |  |
| **13** |  |  |  |
| **14** |  |  |  |
| **15** |  |  |  |
| **16** |  |  |  |
| **17** |  |  |  |
| **18** |  |  |  |
| **19** |  |  |  |
| **20** |  |  |  |

1-a. Write a C program to find out the sum all even elements from an array.

#include <stdio.h>

int main()

{

int arr[50], n, sum = 0, i;

printf("Enter no of elements of array:\t");

scanf("%d", &n);

printf("Enter elements:\n");

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

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

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

{

if(arr[i]%2 == 0)

sum += arr[i];

}

printf("Sum of even elements is %d", sum);

return 0;

}

Output:

Enter no of elements of array: 5

Enter elements:

1

2

3

4

5

Sum of even elements is 6

1. b. Write a C program to find out union of two given arrays.

#include <stdio.h>

void sort(int \*arr, int n)

{

int i, j, temp;

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

{

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

{

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

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

}

int main()

{

int arr1[50], arr2[50], uArr[100], n1, n2, i = 0, j = 0, k = 0, flag = 0;

printf("Enter no of elements for array 1:\t");

scanf("%d", &n1);

printf("Enter no of elements for array 2:\t");

scanf("%d", &n2);

printf("Enter elements for array 1:\n");

for(i=0;i<n1;i++)

scanf("%d", &arr1[i]);

printf("Enter elements for array 2:\n");

for(i=0;i<n2;i++)

scanf("%d", &arr2[i]);

sort(arr1, n1);

sort(arr2, n2);

i = j = 0;

while(i<n1 && j<n2)

{

if(arr1[i] < arr2[j])

uArr[k++] = arr1[i++];

else if(arr1[i] < arr2[j])

uArr[k++] = arr2[j++];

else

uArr[k++] = arr2[j++]; i++;

}

while(i<n1)

uArr[k++] = arr1[i++];

while(j<n2)

uArr[k++] = arr2[j++];

printf("Union of the two arrays:\n");

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

printf("%d\t", uArr[i]);

return 0;

}

Output:

Enter no of elements for array 1: 4

Enter no of elements for array 2: 3

Enter elements for array 1:

1

5

2

7

Enter elements for array 2:

2

1

20

Union of the two arrays:

1 2 5 20

1. c. Write a C program to find out intersection of two given arrays.

#include <stdio.h>

void sort(int \*arr, int n)

{

int i, j, temp;

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

{

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

{

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

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

}

int main()

{

int arr1[10], arr2[10], iArr[20], i, k = 0;

int n1, n2, j;

printf("Enter no of elements for array 1:\t");

scanf("%d", &n1);

printf("Enter no of elements for array 2:\t");

scanf("%d", &n2);

printf("Enter elements for array 1:\n");

for(i=0;i<n1;i++)

scanf("%d", &arr1[i]);

printf("Enter elements for array 2:\n");

for(i=0;i<n2;i++)

scanf("%d", &arr2[i]);

sort(arr1, n1);

sort(arr2, n2);

i = 0;

j = 0;

while (i < n1 && j < n2)

{

if (arr1[i] < arr2[j])

i++;

else if (arr2[j] < arr1[i])

j++;

else

{

iArr[k++] = arr1[i];

i++;

j++;

}

}

printf("Intersection of the two arrays is:\t");

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

printf("%d\t", iArr[i]);

return 0;

}

Output:

Enter no of elements for array 1: 4

Enter no of elements for array 2: 4

Enter elements for array 1:

1

2

3

4

Enter elements for array 2:

3

4

5

6

Intersection of the two arrays is: 3 4

1-d. Write a C program to store N elements into the array and the reverse the

contents of that array.

#include <stdio.h>

int main()

{

int arr[10], n, i, j, temp;

printf("Enter no. of elements in array:\t");

scanf("%d", &n);

printf("Enter elements of array:\n");

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

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

for(i=0,j=n-1;i<j;i++,j--)

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

printf("Reverse array:\n");

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

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

return 0;

}

Output:

Enter no. of elements in array: 4

Enter elements of array:

1

2

3

4

Reverse array:

4 3 2 1

1. a. Write a C program to find element with maximum value from an array.

#include <stdio.h>

int main()

{

int arr[10], n, i, big;

printf("Enter no. of elements in array:\t");

scanf("%d", &n);

printf("Enter elements of array:\n");

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

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

big = arr[0];

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

{

if(arr[i]>big)

big = arr[i];

}

printf("The biggest element in the array is %d", big);

return 0;

}

Output:

Enter no. of elements in array: 5

Enter elements of array:

1

12

6

468

65

The biggest element in the array is 468

2-b. Write a C program to create a dynamic array.

#include <stdio.h>

#include <stdlib.h>

int main()

{

int \*arr, n, i;

printf("Enter size of array to make:\t");

scanf("%d", &n);

arr = (int \*)malloc(n\*sizeof(int));

printf("Enter elements to insert into dynamic array:\n");

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

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

printf("Dynamic array created");

printf("Contents of dynamic array:\t");

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

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

return 0;

}

Output:

Enter size of array to make: 5

Enter elements to insert into dynamic array:

1

2

3

4

5

Dynamic array createdContents of dynamic array: 1 2 3 4 5

2-c. Write a C program to Implementation Stack Using Array.

#include <stdio.h>

#define MAX 5

void push(int \*stack, int \*front, int item)

{

if(\*front == MAX-1)

{

printf("\nStack Overflow!\n");

return;

}

if(\*front == -1)

{

\*front = 0;

stack[\*front] = item;

}

else

stack[++(\*front)] = item;

printf("\nItem pushed into stack successfully!\n");

}

void pop(int \*stack, int \*front)

{

if(\*front == -1)

{

printf("\nStack Underflow!\n");

return;

}

(\*front)--;

printf("\nItem popped successfully\n");

}

void display(int \*stack, int front)

{

int i;

if (front == -1)

{

printf("\nStack is empty\n");

return;

}

printf("\nStack:\t");

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

printf("%d\t", stack[i]);

printf("\n");

}

int main()

{

int stack[MAX], front = -1, flag = 0, choice, arg;

while(flag == 0)

{

printf("\nChoose an operation to perform:\n");

printf("1- Push an item\n2- Pop an item\n3- Display current stack\nPress 0 to quit\n");

scanf("%d", &choice);

switch (choice)

{

case 1:

printf("Enter an item to push into the stack:\t");

scanf("%d", &arg);

push(stack, &front, arg);

break;

case 2:

pop(stack, &front);

break;

case 3:

display(stack, front);

break;

case 0:

flag = 1;

break;

default:

printf("Wrong input! Try again");

}

}

}

Output:

Choose an operation to perform:

1- Push an item

2- Pop an item

3- Display current stack

Press 0 to quit

1

Enter an item to push into the stack: 5

Item pushed into stack successfully!

Choose an operation to perform:

1- Push an item

2- Pop an item

3- Display current stack

Press 0 to quit

1

Enter an item to push into the stack: 10

Item pushed into stack successfully!

Choose an operation to perform:

1- Push an item

2- Pop an item

3- Display current stack

Press 0 to quit

3

Stack: 5 10

Choose an operation to perform:

1- Push an item

2- Pop an item

3- Display current stack

Press 0 to quit

2

Item popped successfully

Choose an operation to perform:

1- Push an item

2- Pop an item

3- Display current stack

Press 0 to quit

3

Stack: 5

Choose an operation to perform:

1- Push an item

2- Pop an item

3- Display current stack

Press 0 to quit

0

1. a. Write a C program to Implementation queue Using Array.

#include <stdio.h>

#define MAX 5

void insert(int \*queue, int \*front, int \*rear, int item)

{

if(\*rear == MAX-1)

{

printf("Queue is full\n");

return;

}

if(\*front == -1)

{

\*front = 0;

\*rear = 0;

}

else

\*(rear)++;

queue[\*rear] = item;

printf("\nItem inserted successfully!\n");

}

void delete(int \*queue, int \*front, int \*rear)

{

if(\*front == -1)

{

printf("Queue is empty! Can't delete item\n");

return;

}

if(\*front == \*rear)

\*front = -1;

else

(\*front)++;

printf("\nItem removed successfully\n");

}

void display(int \*queue, int front, int rear)

{

int i;

if(front == -1)

{

printf("Queue is empty\n");

return;

}

printf("Queue:\t");

for(i=front;i<=rear;i++)

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

printf("\n");

}

int main()

{

int queue[MAX], front = -1, rear = -1, choice, arg, flag = 0;

while(flag == 0)

{

printf("\nChoose an operation to perform:\n");

printf("1- Insert an item into queue\n");

printf("2- Remove an item from queue\n");

printf("3- Display the queue\n");

printf("Press 0 to quit\n");

scanf("%d", &choice);

switch (choice)

{

case 1:

printf("Enter the item to insert:\t");

scanf("%d", &arg);

insert(queue, &front, &rear, arg);

break;

case 2:

delete(queue, &front, &rear);

break;

case 3:

display(queue, front, rear);

break;

case 0:

flag = 1;

break;

default:

printf("Wrong Input! Try again\n");

}

}

}

Output:

Choose an operation to perform:

1- Insert an item into queue

2- Remove an item from queue

3- Display the queue

Press 0 to quit

1

Enter the item to insert: 5

Item inserted successfully!

Choose an operation to perform:

1- Insert an item into queue

2- Remove an item from queue

3- Display the queue

Press 0 to quit

3

Queue: 5

Choose an operation to perform:

1- Insert an item into queue

2- Remove an item from queue

3- Display the queue

Press 0 to quit

2

Item removed successfully

Choose an operation to perform:

1- Insert an item into queue

2- Remove an item from queue

3- Display the queue

Press 0 to quit

3

Queue is empty

Choose an operation to perform:

1- Insert an item into queue

2- Remove an item from queue

3- Display the queue

Press 0 to quit

0

3-b. Write a C program to convert infix expression into postfix expression.