

DATE:-22/09/22

COURSE NAME:-DATA STRUCTURES FOR EXPRESSION EVALUATION

COURSE CODE:-CSA0374

NAME OF THE STUDENT:-USHA NANDHINI.B

REGNO:-192111159

EXPERIMENT:-1(MATRIX OUT PUT)

The screenshot shows the Dev-C++ IDE interface with the file '1ST.cpp' open. The code implements matrix multiplication of two 3x3 matrices. The user inputs row=2 and column=2. The program then prompts for the first matrix elements, followed by the second matrix elements. The output shows the resulting 3x3 matrix.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int a[10][10],b[10][10],mul[10][10],r,c,i,j,k;
    system("cls");
    printf("enter the number of row=");
    scanf("%d",&r);
    printf("enter the number of column=");
    scanf("%d",&c);
    printf("enter the first matrix element=\n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("enter the second matrix element=\n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            scanf("%d",&b[i][j]);
        }
    }
}
```

enter the number of row=2
enter the number of column=2
enter the first matrix element=
1
1
2
2
enter the second matrix element=
1
1
2
2
multiply of the matrix=
3 3
6 6

Process exited after 18.43 seconds with return value 0
Press any key to continue . . .

EXPERIMENT:-2(EVEN ODD OUTPUT)

The screenshot shows the Dev-C++ IDE interface with the file '1ST.cpp' open. The code is identical to Experiment 1, performing matrix multiplication of two 3x3 matrices. The user inputs row=2 and column=2. The program prompts for matrix elements and outputs the result.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int a[10][10],b[10][10],mul[10][10],r,c,i,j,k;
    system("cls");
    printf("enter the number of row=");
    scanf("%d",&r);
    printf("enter the number of column=");
    scanf("%d",&c);
    printf("enter the first matrix element=\n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("enter the second matrix element=\n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            scanf("%d",&b[i][j]);
        }
    }
}
```

enter the number of row=2
enter the number of column=2
enter the first matrix element=
1
1
2
2
enter the second matrix element=
1
1
2
2
multiply of the matrix=
3 3
6 6

Process exited after 18.43 seconds with return value 0
Press any key to continue . . .

EXPERIMENT:-3(FACTORIAL WITHOUT OUTPUT)

C:\Users\chall\OneDrive\Documents\fatorial.cpp - [Executing] - Dev-C++ 5.11

```

File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug factorial.cpp
1 #include <stdio.h>
2 int main() {
3     int n, i;
4     unsigned long long fact = 1;
5     printf("Enter an integer: ");
6     scanf("%d", &n);
7
8     // shows error if the user enters a negative integer
9     if (n < 0)
10        printf("Error! Factorial of a negative number doesn't exist.");
11    else {
12        for (i = 1; i <= n; ++i) {
13            fact *= i;
14        }
15        printf("Factorial of %d = %llu", n, fact);
16    }
17
18    return 0;
19

```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation C:\Users\chall\OneDrive\Documents\fatorial.exe

Enter an integer: 5
Factorial of 5 = 120

Process exited after 5.622 seconds with return value 0
Press any key to continue . . .

Line: 19 Col: 1331 14 19-09-2022

92°F Mostly sunny

EXPERIMENT:4(FIBONACCI WITHOUT OUTPUT)

C:\Users\chall\OneDrive\Documents\using fac.cpp - [Executing] - Dev-C++ 5.11

```

File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug using fac.cpp
1 #include<stdio.h>
2 int main()
3 {
4     int n1=0,n2=1,n3,i,number;
5     printf("Enter the number of elements:");
6     scanf("%d",&number);
7     printf("\n%d %d",n1,n2);//printing 0 and 1
8     for(i=2;i<number;i++)//Loop starts from 2 because 0 and 1 are already printed
9     {
10         n3=n1+n2;
11         printf(" %d",n3);
12         n1=n2;
13         n2=n3;
14     }
15     return 0;
16

```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation C:\Users\chall\OneDrive\Documents\using fac.exe

Enter the number of elements:3
0 1 1

Process exited after 6.864 seconds with return value 0
Press any key to continue . . .

Output Size: 126.1015625 KB
Compilation Time: 1.44s

Line: 16 Col: 3 Sel: 0 Lines: 16 Length: 389 Insert Done parsing in 0.016 seconds 19:57 17 19-09-2022

EXPERIMENT:5(FACTORIAL USING OUTPUT)

C:\Users\chall\OneDrive\Documents\using fac.cpp - Dev-C++ 5.11

The screenshot shows the Dev-C++ IDE interface. The code editor displays a C++ program named 'fac.cpp' that calculates the factorial of a given integer. The output window shows the execution of the program, prompting for input and displaying the result. The taskbar at the bottom shows various application icons.

```

1 #include<stdio.h>
2 long int multiplyNumbers(int n);
3 int main() {
4     int n;
5     printf("Enter a positive integer: ");
6     scanf("%d",&n);
7     printf("Factorial of %d = %ld", n, multiplyNumbers(n));
8     return 0;
9 }
10 long int multiplyNumbers(int n) {
11     if (n>1)
12         | return n*multiplyNumbers(n-1);
13     else
14         | return 1;
15 }
16

```

Select C:\Users\chall\OneDrive\Documents\using fac.exe
Compiler Enter a positive integer: 8 Factorial of 8 = 40320 Abort Compilation Process exited after 3.288 seconds with return value 0 Press any key to continue . . .

Line: 4 Col: 11 Sel: 0 Lines: 16 Length: 346 Insert Done parsing in 0.063 seconds

19:50 19-09-2022

EXPERIMENT:6(FIBONACCI USING OUTPUT)

C:\Users\chall\OneDrive\Documents\using fac.cpp - [Executing] - Dev-C++ 5.11

The screenshot shows the Dev-C++ IDE interface. The code editor displays a C++ program named 'fac.cpp' that prints a Fibonacci series. The output window shows the execution of the program, prompting for input and displaying the series. The taskbar at the bottom shows various application icons.

```

1 #include<stdio.h>
2 void printfibonacci(int n){
3     static int n1=0,n2=1,n3;
4     if(n>0){
5         | n3 = n1 + n2;
6         | n1 = n2;
7         | n2 = n3;
8         | printf("%d ",n3);
9         | printfibonacci(n-1);
10    }
11 }
12 int main(){
13     int n;
14     printf("enter the number of elements: ");
15     scanf("%d",&n);
16     printf("fibonacci series: ");
17     printf("%d %d ",0,1);
18     printfibonacci(n-2); //n-2 because 2 numbers are already printed
19     return 0;
20 }

```

Select C:\Users\chall\OneDrive\Documents\using fac.exe
Compiler enter the number of elements: 10 Fibonacci series: 0 1 1 2 3 5 8 13 21 34 Abort Compilation Process exited after 6.394 seconds with return value 0 Press any key to continue . . .

Line: 1 Col: 2 Sel: 0 Lines: 20 Length: 537 Insert Done parsing in 0.016 seconds

19:53 19-09-2022

EXPERIMENT:-7(ARRAY OUTPUT)

The screenshot shows the Dev-C++ IDE interface. The main window displays the code for 'linear search.cpp'. The code implements a linear search algorithm to find a key in an array. It includes a function 'findelement' and a main function that prints the position of the key or a message if it's not found. The output window shows the execution of the program, which finds the key at position 9. The status bar at the bottom provides system information like weather and date.

```

1 #include<stdio.h>
2 int findelement(int a[],int n,int key){
3     int i;
4     for( i=0;i<n;++i)
5     {
6         if (a[i]==key)
7             return i;
8     }
9     return -1;
10}
11int main(){
12    int a[]={3,5,7,9,8,22};
13    int n(sizeof(a)/sizeof(a[0]));
14    int key=9;
15    int position=findelement(a,n,key);
16    if(position==-1)
17    {
18        printf("elements %d notfound",key);
19    }
20    else
21    {
22        printf("position of %d:%d",key,position+9);
23    }
24    return 0;
25}

```

ARRAY INSERTION(OUTPUT)

The screenshot shows the Dev-C++ IDE interface. The main window displays the code for 'array insert.cpp'. The code prompts the user to enter values into an array, specifies a location for insertion, and then shifts elements to make space for the new value. The output window shows the user entering values [3, 6, 8, 9] and inserting the value 5 at index 3, resulting in the array [5, 6, 8, 9]. The status bar at the bottom provides system information like weather and date.

```

1 #include<stdio.h>
2 int main()
3 {
4     int position,c,n,value,array[50];
5     printf("enter the number of values in an array");
6     scanf("%d",&n);
7     printf("enter the values of %d\n",n);
8     for(c=0;<n;c++)
9     {
10        scanf("%d",&array[c]);
11        printf("enter the location");
12        scanf("%d",&position);
13        printf("enter the value");
14        scanf("%d",&value);
15        for(c=n-1;c>=position-1;c--)
16            array[c+1]=array[c];
17        array[position-1]=value;
18        printf("resultant array is");
19        for(c=0;<n;c++)

```

ARRAY DELETION(OUTPUT)

The screenshot shows the Dev-C++ IDE interface. The code in the editor is:

```
10 for (i = 0; i < num; i++)
11 {
12     printf (" arr[%d] = ", i);
13     scanf ("%d", &arr[i]);
14 }
15 printf( " Define the position of the array element where you
16 scanf (" %d", &pos);
17 if (pos >= num+1)
18 {
19     printf (" \n Deletion is not possible in the array.");
20 }
21 else
22 {
23     for (i = pos - 1; i < num -1; i++)
24     {
25         arr[i] = arr[i+1];
26     }
27     printf (" \n The resultant array is: \n");
28     for (i = 0; i< num - 1; i++)
29     {
30         printf (" arr[%d] = ", i);
31         printf (" %d \n", arr[i]);
32     }
33 }
34 }
```

The output window shows the execution of the program:

```
Select C:\Users\chall\OneDrive\Documents\array insert.exe
Enter the number of elements in an array: 5
Enter 5 elements in array:
arr[0] = 5
arr[1] = 7
arr[2] = 8
arr[3] = 9
arr[4] = 2
Define the position of the array element where you want to delete
2

The resultant array is:
arr[0] = 5
arr[1] = 8
arr[2] = 9
arr[3] = 2
```

Process exited after 31.25 seconds with return value 0
Press any key to continue . . .

ARRAY DISPLAY(OUTPUT)

The screenshot shows the Dev-C++ IDE interface. The code in the editor is:

```
1 #include <stdio.h>
2 void display(int age1, int age2) {
3     printf("%d\n", age1);
4     printf("%d\n", age2);
5 }
6 int main() {
7     int ageArray[] = {2, 8, 4, 12};
8     display(ageArray[1], ageArray[2]);
9     return 0;
10 }
```

The output window shows the execution of the program:

```
C:\Users\chall\OneDrive\Documents\array insert.exe
-----
Process exited after 0.04579 seconds with return value 0
Press any key to continue . . .
```

EXPERIMENT:-8(LINEARSEARCH(OUTPUT))

```

1 #include<stdio.h>
2 int linearsearch(int a[],int n,int val){
3     for(int i=0;i<n;i++)
4     {
5         if (a[i]==val)
6             return i+1;
7     }
8     return -1;
9 }
10 int main(){
11     int a[]={3,5,7,9,8,22};
12     int val=21;
13     int n(sizeof(a)/sizeof(a[0]));
14     int res=linearsearch(a,n,val);
15     printf("the elements of the array are:");
16     for(int i=0;i<n;i++)
17         printf("%d",a[i]);
18     printf("\nelements to be searched is-%d",val);
19     if(res==1)
20         printf("\nelements is not present in the array:");
21     else
22         printf("\nelements is presents at %d position of array",res);
23     return 0;
24 }
25

```

Select C:\Users\chall\OneDrive\Documents\linear search.exe
the elements of the array are:3579822
elements to be searched is-21
elements is not present in the array:

Process exited after 0.08195 seconds with return value 0
Press any key to continue . . .

EXPERIMENT:9(BINARYSEARCH(OUTPUT))

```

1 #include <stdio.h>
2 int binarySearch(int array[], int x, int low, int high) {
3     // Repeat until the pointers low and high meet each other
4     while (low <= high) {
5         int mid = low + (high - low) / 2;
6         if (array[mid] == x)
7             return mid;
8         if (array[mid] < x)
9             low = mid + 1;
10        else
11            high = mid - 1;
12    }
13    return -1;
14 }
15 int main(void) {
16     int array[] = {3, 4, 5, 6, 7, 8, 9};
17     int n = sizeof(array) / sizeof(array[0]);
18     int x = 4;
19     int result = binarySearch(array, x, 0, n - 1);
20     if (result == -1)
21         printf("Not found");
22     else
23         printf("Element is found at index %d", result);
24     return 0;
25 }

```

C:\Users\chall\OneDrive\Documents\linear search.exe
Element is found at index 1

Process exited after 0.06617 seconds with return value 0
Press any key to continue . . .

EXPERIMENT:10(LINKED LIST(OUTPUT))

```

C:\Users\chall\OneDrive\Documents\array insert.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug [*] array insert.cpp
79 void printList(struct Node* node) {
80     while (node != NULL) {
81         printf("%d ", node->data);
82         node = node->next;
83     }
84 }
85 int main() {
86     struct Node* head = NULL;
87     insertAtEnd(&head, 1);
88     insertAtBeginning(&head, 2);
89     insertAtBeginning(&head, 3);
90     insertAtEnd(&head, 4);
91     insertAfter(head->next, 5);
92     printf("Linked list: ");
93     printList(head);
94     printf("\nAfter deleting an element: ");
95     deleteNode(&head, 3);
96     printList(head);
97     int item_to_find = 3;
98     if (searchNode(&head, item_to_find)) {
99         printf("\n%d is found", item_to_find);
100    } else {
101        printf("\n%d is not found", item_to_find);
102    }
103    sortLinkedList(&head);
104    printf("\nSorted List: ");

```

Compiler Resources Compile Log Debug Find Results

Line: 79 Col: 3 Sel: 0 Lines: 106 Length: 2754 Insert Done parsing

90°F Mostly cloudy

C:\Users\chall\OneDrive\Documents\array insert.exe

Linked list: 3 2 5 1 4
After deleting an element: 2 5 1 4
3 is not found
Sorted List: 1 2 4 5

Process exited after 0.05659 seconds with return value 0
Press any key to continue . . .

ENG IN 16:50 20-09-2022

EXPERIMENT:11(STACK OPERATIONS(OUTPUT))

```

C:\Users\chall\OneDrive\Documents\tree tran.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug tree tran.cpp
1 #include <stdio.h>
2 void TOH(int n,char x,char y,char z){
3     if(n>0){
4         TOH(n-1,x,z,y);
5         printf("\n%c to %c",x,y);
6         TOH(n-1,z,y,x);
7     }
8 }
9 int main()
10 {
11     int n=3;
12     TOH(n,'A','B','C');
13 }

```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\chall\OneDrive\Documents\tree tran.
- Output Size: 128.4598609375 KiB
- Compilation Time: 0.31s

Line: 7 Col: 6 Sel: 0 Lines: 13 Length: 195 Insert Done parsing in 0 seconds

92°F Mostly sunny

C:\Users\chall\OneDrive\Documents\tree tran.exe

A to B
A to C
B to C
A to B
C to A
C to B
A to B

Process exited after 0.0722 seconds with return value 0
Press any key to continue . . .

ENG IN 14:52 21-09-2022

EXPERIMENT:12(STACK APPLICATION)

C:\Users\chall\OneDrive\Documents\tree tran.cpp - [Executing] - Dev-C++ 5.11

```

1 #include <stdio.h>
2 void TOH(int n,char x,char y,char z){
3     if(n>0){
4         TOH(n-1,x,z,y);
5         printf("\n%c to %c",x,y);
6         TOH(n-1,z,y,x);
7     }
8 }
9 int main()
10 {
11     int n=3;
12     TOH(n,'A','B','C');
13 }

```

C:\Users\chall\OneDrive\Documents\tree tran.exe

A to B
A to C
B to C
A to B
C to A
C to B
B to A

Process exited after 0.0722 seconds with return value 0
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\chall\OneDrive\Documents\tree tran.
- Output Size: 120.4599609375 KiB
- Compilation Time: 0.31s

Line: 7 Col: 6 Sel: 0 Lines: 13 Length: 195 Insert Done parsing in 0 seconds

92°F Mostly sunny

EXPERIMENT:13(QUEUE OPERATIONS(OUTPUT))

C:\Users\chall\OneDrive\Documents\hashing lin.cpp - [Executing] - Dev-C++ 5.11

```

46 {
47     if(front===-1)
48     {
49         printf("\nqueue is empty!!!");
50     }
51     else
52     {
53         printf("\nDeleted element is %d",queue[rear]);
54         front=front-1;
55     }
56 }
57 void display()
58 {
59     int i;
60     if(front===-1)
61     {
62         printf("\nqueue is empty!!!");
63     }
64     else
65     {
66         printf("\nqueue is...\\n");
67         for(i=front;i>=0;i--)
68             printf("%d\\n",queue[i]);
69     }
70 }

```

Select C:\Users\chall\OneDrive\Documents\hashing lin.exe

Enter your choice(1-4):1

Enter element to enqueue:4

1.enqueue
2.dequeue
3.Display
4.Exit

Enter your choice(1-4):2

Deleted element is 0

1.enqueue
2.dequeue
3.Display
4.Exit

Enter your choice(1-4):3

queue is empty!!

1.enqueue
2.dequeue
3.Display
4.Exit

Enter your choice(1-4):1

94°F Mostly sunny

EXPERIMENT:14(TREE TRANSVERSAL(OUTPUT))

The screenshot shows the Dev-C++ IDE interface. The left pane displays the source code for `tree tran.cpp`. The right pane shows the terminal window output of the program `tree tran.exe`. The output shows three types of tree traversals:

```

Preorder traversal of binary tree is
1 2 4 5 3
Inorder traversal of binary tree is
4 2 5 1 3
Postorder traversal of binary tree is
4 5 2 3 1

```

The code implements three traversal methods: Preorder, Inorder, and Postorder. It also includes a main function that creates a binary tree and prints its Preorder traversal.

EXPERIMENT:15(HASH USING LINEAR PROBING(OUTPUT))

The screenshot shows the Dev-C++ IDE interface. The left pane displays the source code for `hashing lin.cpp`. The right pane shows the terminal window output of the program `hashing lin.exe`. The output demonstrates the use of linear probing for a hash table of size 10. It includes options for Insert, Display, Search, and Exit, and shows the state of the hash table and search results.

```

1
enter a value to insert into hash table
22

Press 1. Insert 2. Display 3. Search 4.Exit
2

elements in the hash table are

at index 0      value =  0
at index 1      value =  0
at index 2      value =  22
at index 3      value =  0
at index 4      value =  0
at index 5      value =  0
at index 6      value =  0
at index 7      value =  0
at index 8      value =  8
at index 9      value =  0
Press 1. Insert 2. Display 3. Search 4.Exit
3

enter search element
42

value is not found
Press 1. Insert 2. Display 3. Search 4.Exit

```

EXPERIMENT:16(INSERTION SORTING(OUTPUT))

The screenshot shows the Dev-C++ IDE interface. On the left, the code editor displays a C++ file named 'hashing lin.cpp' containing a merge sort algorithm. On the right, the terminal window shows the execution of the program. The user enters 5 as the number of elements and the numbers 80, 7, 121, and 3. The sorted output is 378098121. The status bar at the bottom shows the date as 21-09-2022.

```
1 #include<stdio.h>
2 int main()
3 {
4     int i,j,count,temp,num[25];
5     printf("how many numbers u are going to enter?:");
6     scanf("%d",&count);
7     printf("enter number of elements:");
8     for(i=0;i<count;i++)
9     {
10         scanf("%d",&num[i]);
11     }
12     for(i=1;i<count;i++)
13     {
14         temp=num[i];
15         j=i-1;
16         while((temp<num[j])&&(j>=0))
17         {
18             num[j+1]=num[j];
19             j=j-1;
20         }
21         num[j+1]=temp;
22     }
23     printf("order of sorted elements:");
24     for(i=0;i<count;i++)
25     {
26         printf("%d",num[i]);
27     }
}
```

```
how many numbers u are going to enter?:5
enter number of elements:98
80
7
121
3
order of sorted elements:378098121
-----
Process exited after 24.35 seconds with return value 0
Press any key to continue . . .
```

EXPERIMENT:17(MERGE SORTING)

The screenshot shows a Windows terminal window titled 'C:\Users\ASUS\OneDrive\Documents\merge.exe'. It displays the list before sorting (10, 14, 19, 26, 27, 31, 33, 35, 42, 44, 8) and the list after sorting (8, 10, 14, 19, 26, 27, 31, 33, 35, 42, 44). The status bar at the bottom shows the date as 22-09-2022.

```
List before sorting
10 14 19 26 27 31 33 35 42 44 8
List after sorting
8 10 14 19 26 27 31 33 35 42 44
-----
Process exited after 0.6279 seconds with return value 3
Press any key to continue . . .
```

EXPERIMENT:-18(QUICK SORT(OUTPUT))

```
C:\Users\ASUS\OneDrive\Documents\Quick.exe
Enter total no.of elements: 7
Enter the elements: 10
5
25
48
68
28
100
Sorted Array:  5 10 25 28 48 68 100
-----
Process exited after 18.39 seconds with return value 0
Press any key to continue . . .


```

EXPERIMENT:-19(HEAP SORT (OUTPUT))

```
C:\Users\ASUS\OneDrive\Documents\heap.exe
Before sorting array elements are -
48 10 23 43 28 26 1
After sorting array elements are -
1 10 23 26 28 43 48
-----
Process exited after 0.7099 seconds with return value 0
Press any key to continue . . .


```

EXPERIMENT:20(AVL OPERATIONS(OUTPUT))

```
C:\Users\ASUS\OneDrive\Documents\AVL trees.exe
4 2 1 3 7 5 8
After deletion: 4 2 1 7 5 8
-----
Process exited after 0.61 seconds with return value 0
Press any key to continue . . .
```

EXPERIMENT:21(BFS(OUTPUT))

```
C:\Users\ASUS\OneDrive\Documents\Breadth.exe
Enter the number of vertices:4
Enter graph data in matrix form:
1 1 1 1
0 1 0 0
0 0 1 0
0 0 0 1
Enter the starting vertex:1
The node which are reachable are:
1 2 3 4
-----
Process exited after 26.89 seconds with return value 0
Press any key to continue . . .
```

EXPERIMENT:22(DFS(OUTPUT))

```
C:\Users\ASUS\OneDrive\Documents\Depth.exe
Enter no. of vertices:4
Enter the adjacency matrix:
1 1 1 1
1 0 1 0
0 0 0 0
0 0 0 1
Enter the starting node:1
Distance of node0=1
Path0<-1
Distance of node2=1
Path2<-1
Distance of node3=2
Path3<-0<-1
-----
Process exited after 35.38 seconds with return value 0
Press any key to continue . . .
```

EXPERIMENT:23(DIJKSTRA ALGORITHM(OUTPUT))

```
C:\Users\ASUS\OneDrive\Documents\Dijsktra's.exe
Distance from source to 1: 3
Distance from source to 2: 1
Distance from source to 3: 2
Distance from source to 4: 4
Distance from source to 5: 4
Distance from source to 6: 3
-----
Process exited after 0.5575 seconds with return value 0
Press any key to continue . . .
```

EXPERIMENT:24(PRIM'S ALGORITHM(OUTPUT))

```
C:\Users\ASUS\OneDrive\Documents\Prim's.exe
Edge      Weight
3 <-> 1    4
0 <-> 2    3
2 <-> 3    2
3 <-> 4    1

Process exited after 0.6292 seconds with return value 0
Press any key to continue . . .
```

EXPERIMENT:25(KRUKAL ALGORITHM(OUTPUT))

```
C:\Users\ASUS\OneDrive\Documents\Kruskal.exe
2 - 1 : 2
5 - 2 : 2
3 - 2 : 3
4 - 3 : 3
1 - 0 : 4
Spanning tree cost: 14

Process exited after 0.9241 seconds with return value 23
Press any key to continue . . .
```