Day -2

Assignment

Menda Mani Sai

192111399

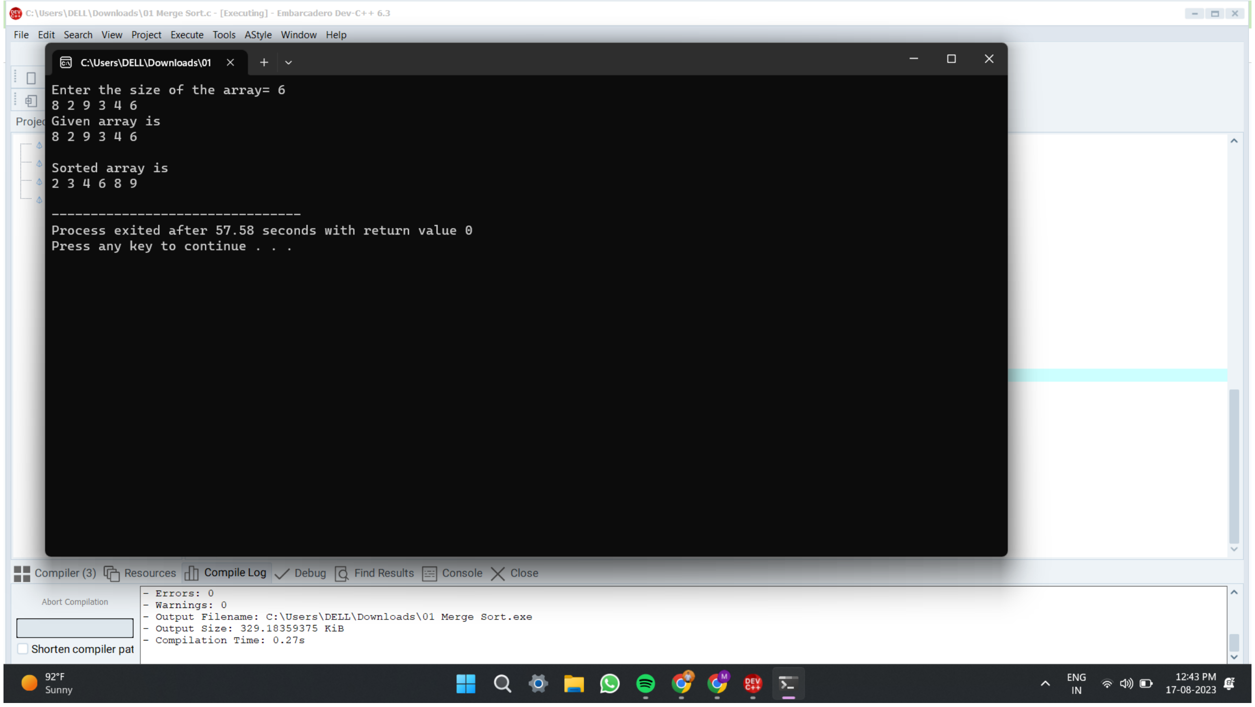
1. Write a program to perform Merge Sort.

```
#include<stdio.h>
#include <stdlib.h>
void merge(int arr[],int l, int m, int r)
{
    int i,j,k;
    int n1=m-l+1;
    int n2=r-m;

    int L[n1],R[n2];

      for (i = 0; i < n1; i++)
                  L[i] = arr[l + i];
         for (j = 0; j < n2; j++)
                  R[j] = arr[m + 1 +j];

    i = 0;
         j = 0;
         k = l;
         while (i < n1 && j < n2) {
                  if (L[i] <= R[j]) {
                           arr[k] = L[i];
                           i++;
                  }
                  else {
                           arr[k] = R[j];
                           j++;
                  }
                  k++;
         }

  while (i < n1) {
```

```
                arr[k] = L[i];
                i++;
                k++;
        }
 while (j < n2) {
                arr[k] = R[j];
                j++;
                k++;
        }
}
  void mergeSort(int arr[], int l, int r)
{
        if (l < r) {
                int m = l + (r - l) / 2;

                mergeSort(arr, l, m);
                mergeSort(arr, m + 1, r);

                merge(arr, l, m, r);
        }
}

void printArray(int a[], int n)
{
        int i;
        for (i = 0; i < n; i++)
                printf("%d ", a[i]);
        printf("\n");
}

int main()
{
  int n;
  printf("Enter the size of the array= ");
  scanf("%d", &n);
  int arr[n];
  for(int i=0;i<n;i++){
    scanf("%d", &arr[i]);
  }
  int arr_n = sizeof(arr) / sizeof(arr[0]);
       printf("Given array is \n");
        printArray(arr, arr_n);

        mergeSort(arr, 0, arr_n - 1);
```

```
          printf("\nSorted array is \n");
          printArray(arr, arr_n);
  return 0;
}
```

OUTPUT:


C:\Users\DELL\Downloads\01 Merge Sort.c - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
C:\Users\DELL\Downloads\01
Compiler (3)
Resources
Compile Log
Debug
Find Results
Console
Close
Abort Compilation
Shorten compiler pat
92°F
Sunny
ENG
IN
12:43 PM
17-08-2023

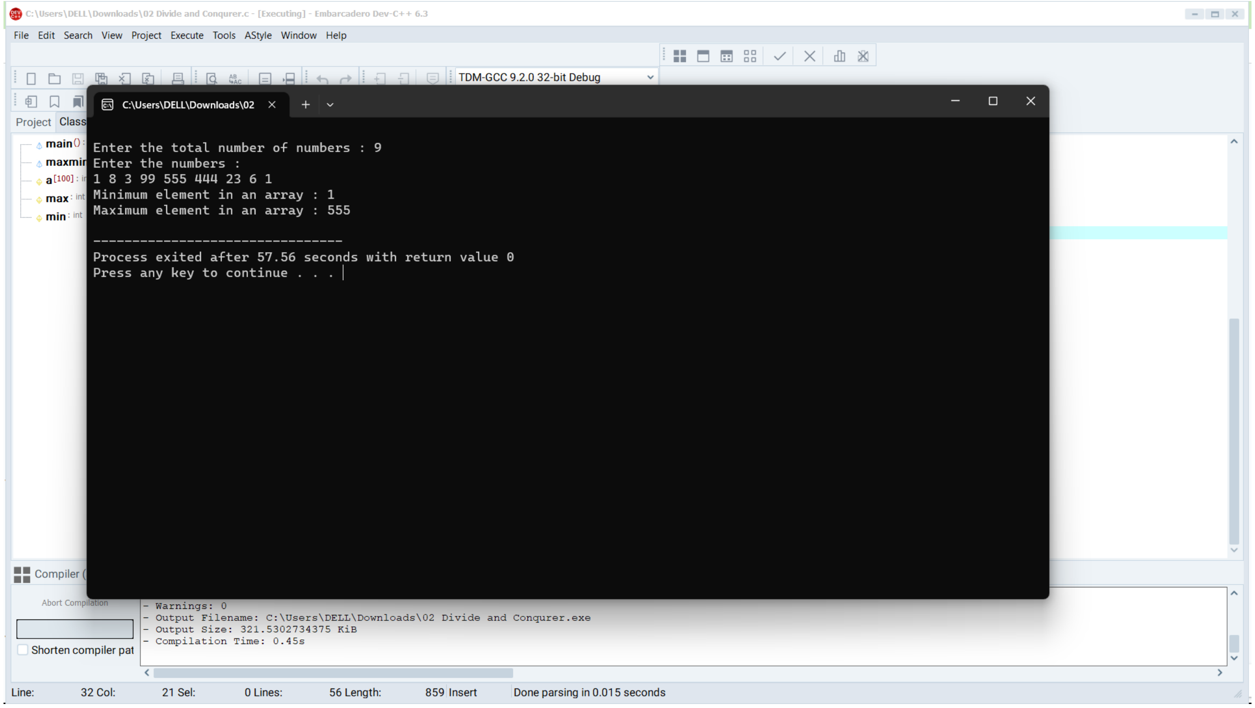
```
Enter the size of the array= 6
8 2 9 3 4 6
Given array is
8 2 9 3 4 6

Sorted array is
2 3 4 6 8 9

--------------------------------
Process exited after 57.58 seconds with return value 0
Press any key to continue . . .
```

```
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\DELL\Downloads\01 Merge Sort.exe
- Output Size: 329.18359375 KiB
- Compilation Time: 0.27s
```

2. Using Divide and Conquer strategy to find Max and Min value in the list.

```
#include<stdio.h>
#include<stdio.h>
int max, min;
int a[100];
void maxmin(int i, int j)
{
 int max1, min1, mid;
 if(i==j)
 {
  max = min = a[i];
 }
 else
 {
  if(i == j-1)
  {
   if(a[i] <a[j])
   {
    max = a[j];
    min = a[i];
   }
   else
   {
    max = a[i];
    min = a[j];
   }
  }
  else
  {
   mid = (i+j)/2;
   maxmin(i, mid);
   max1 = max; min1 = min;
   maxmin(mid+1, j);
   if(max <max1)
    max = max1;
   if(min > min1)
    min = min1;
  }
 }
}
int main ()
{
 int i, num;
 printf ("\nEnter the total number of numbers : ");
 scanf ("%d",&num);
 printf ("Enter the numbers : \n");
 for (i=1;i<=num;i++)
  scanf ("%d",&a[i]);

 max = a[0];
 min = a[0];
 maxmin(1, num);
 printf ("Minimum element in an array : %d\n", min);
 printf ("Maximum element in an array : %d\n", max);
 return 0;
}
```

OUTPUT:



C:\Users\DELL\Downloads\02 Divide and Conqurer.c - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 32-bit Debug
C:\Users\DELL\Downloads\02
Project Class
main()
maxmin
a[100]
max : int
min : int
Compiler
Abort Compilation
Shorten compiler pat

```
Enter the total number of numbers : 9
Enter the numbers :
1 8 3 99 555 444 23 6 1
Minimum element in an array : 1
Maximum element in an array : 555

--------------------------------
Process exited after 57.56 seconds with return value 0
Press any key to continue . . .
```

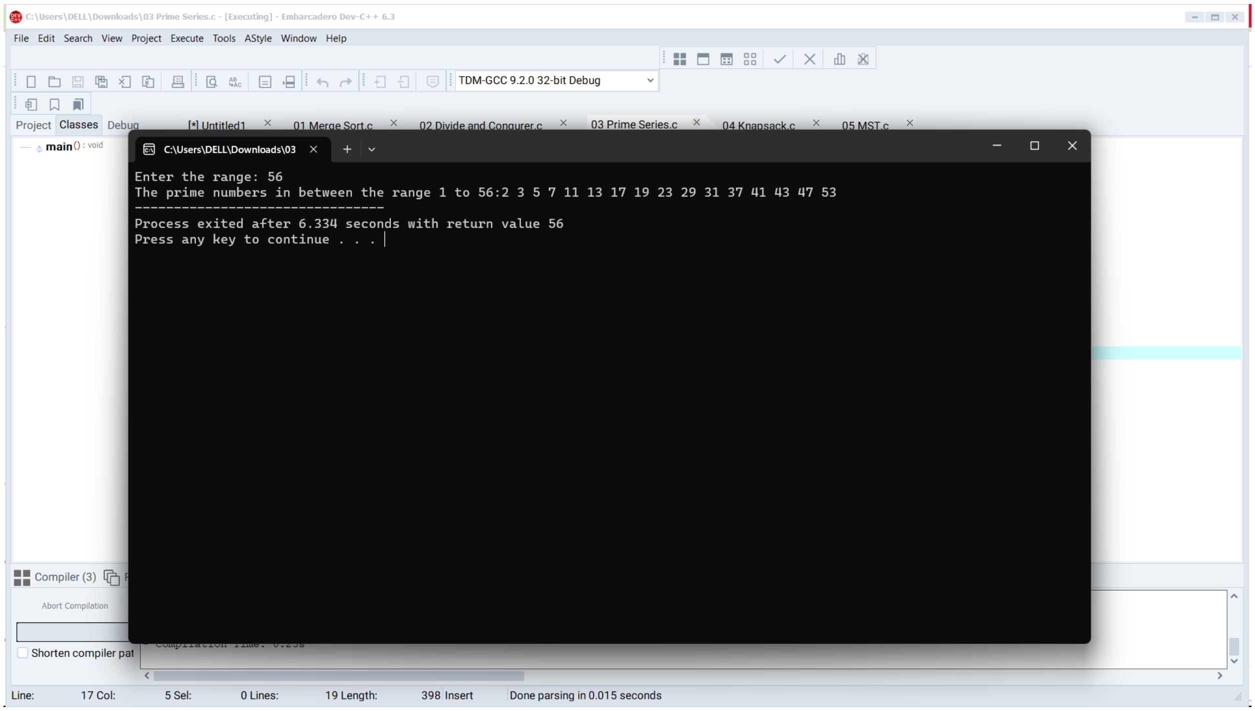
```
- Warnings: 0
- Output Filename: C:\Users\DELL\Downloads\02 Divide and Conqurer.exe
- Output Size: 321.5302734375 KiB
- Compilation Time: 0.45s
```

Line: 32 Col: 21 Sel: 0 Lines: 56 Length: 859 Insert Done parsing in 0.015 seconds

3. Write a program to generate all the prime numbers.

```
#include<stdio.h>
void main(){
  int i, num, n, count;
  printf("Enter the range: ");
  scanf("%d", &n);
  printf("The prime numbers in between the range 1 to %d:",n);
  for(num = 1;num<=n;num++){
    count = 0;
    for(i=2;i<=num/2;i++){
      if(num%i==0){
        count++;
      break;
    }
  }
  if(count==0 && num!= 1)
    printf("%d ",num);
  }
}
```

OUTPUT:


C:\Users\DELL\Downloads\03 Prime Series.c - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 32-bit Debug
Project Classes Debug
[*] Untitled1
01 Merge Sort.c
02 Divide and Conqurer.c
03 Prime Series.c
04 Knapsack.c
05 MST.c
main () : void
C:\Users\DELL\Downloads\03
Compiler (3)
Abort Compilation
Shorten compiler pat
Line: 17 Col: 5 Sel: 0 Lines: 19 Length: 398 Insert Done parsing in 0.015 seconds

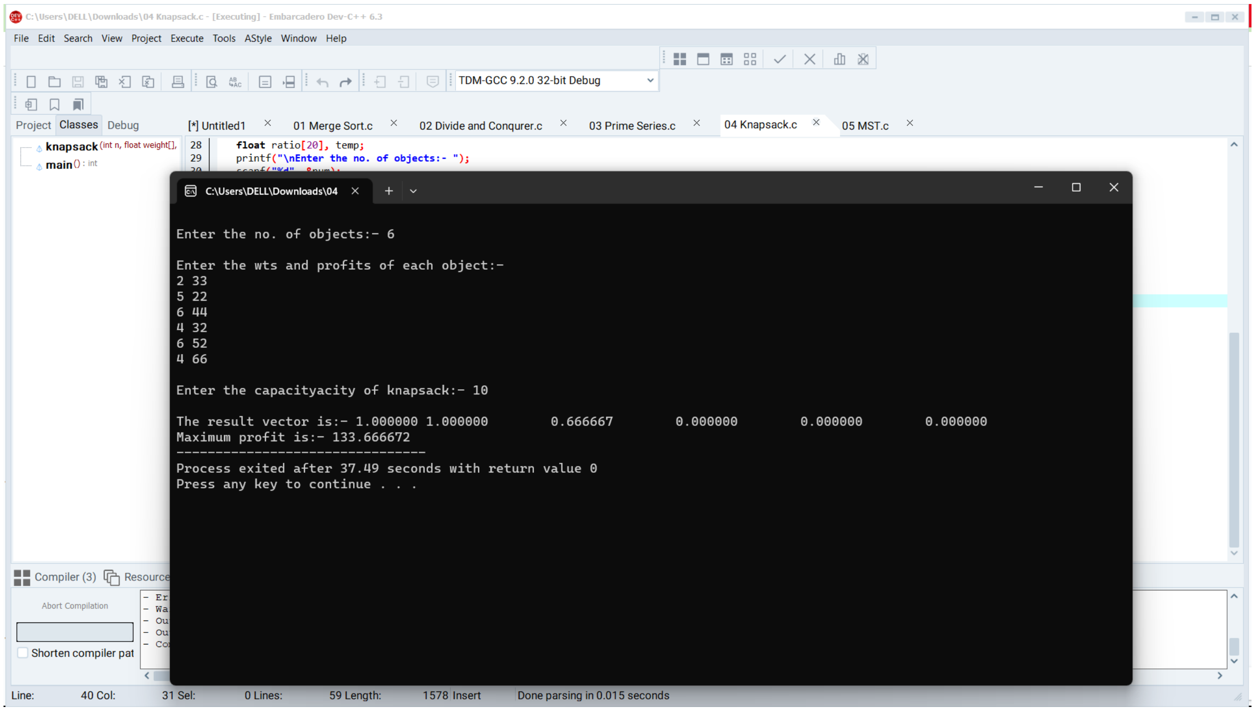
```
Enter the range: 56
The prime numbers in between the range 1 to 56:2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53
--------------------------------
Process exited after 6.334 seconds with return value 56
Press any key to continue . . .
```

4. Write a program to perform Knapsack problem using greedy techniques.

```
# include<stdio.h>
void knapsack(int n, float weight[], float profit[], float capacity) {
  float x[20], tp = 0;
  int i, j, u;
  u = capacity;
  for (i = 0; i < n; i++)
    x[i] = 0.0;
  for (i = 0; i < n; i++) {
    if (weight[i] > u)
      break;
    else {
      x[i] = 1.0;
      tp = tp + profit[i];
      u = u - weight[i];
    }
```

```
   }
   if (i < n)
      x[i] = u / weight[i];
   tp = tp + (x[i] * profit[i]);
   printf("\nThe result vector is:- ");
   for (i = 0; i < n; i++)
      printf("%f\t", x[i]);
   printf("\nMaximum profit is:- %f", tp);
}
int main() {
   float weight[20], profit[20], capacity;
   int num, i, j;
   float ratio[20], temp;
   printf("\nEnter the no. of objects:- ");
   scanf("%d", &num);
   printf("\nEnter the wts and profits of each object:- ");
   for (i = 0; i < num; i++) {
      scanf("%f %f", &weight[i], &profit[i]);
   }
   printf("\nEnter the capacityacity of knapsack:- ");
   scanf("%f", &capacity);
   for (i = 0; i < num; i++) {
      ratio[i] = profit[i] / weight[i];
   }
   for (i = 0; i < num; i++) {
      for (j = i + 1; j < num; j++) {
         if (ratio[i] < ratio[j]) {
            temp = ratio[j];
            ratio[j] = ratio[i];
            ratio[i] = temp;
            temp = weight[j];
            weight[j] = weight[i];
            weight[i] = temp;
            temp = profit[j];
            profit[j] = profit[i];
            profit[i] = temp;
         }
      }
   }

   knapsack(num, weight, profit, capacity);
   return(0);
}
```

OUTPUT:



```
    float ratio[20], temp;
    printf("\nEnter the no. of objects:- ");
```

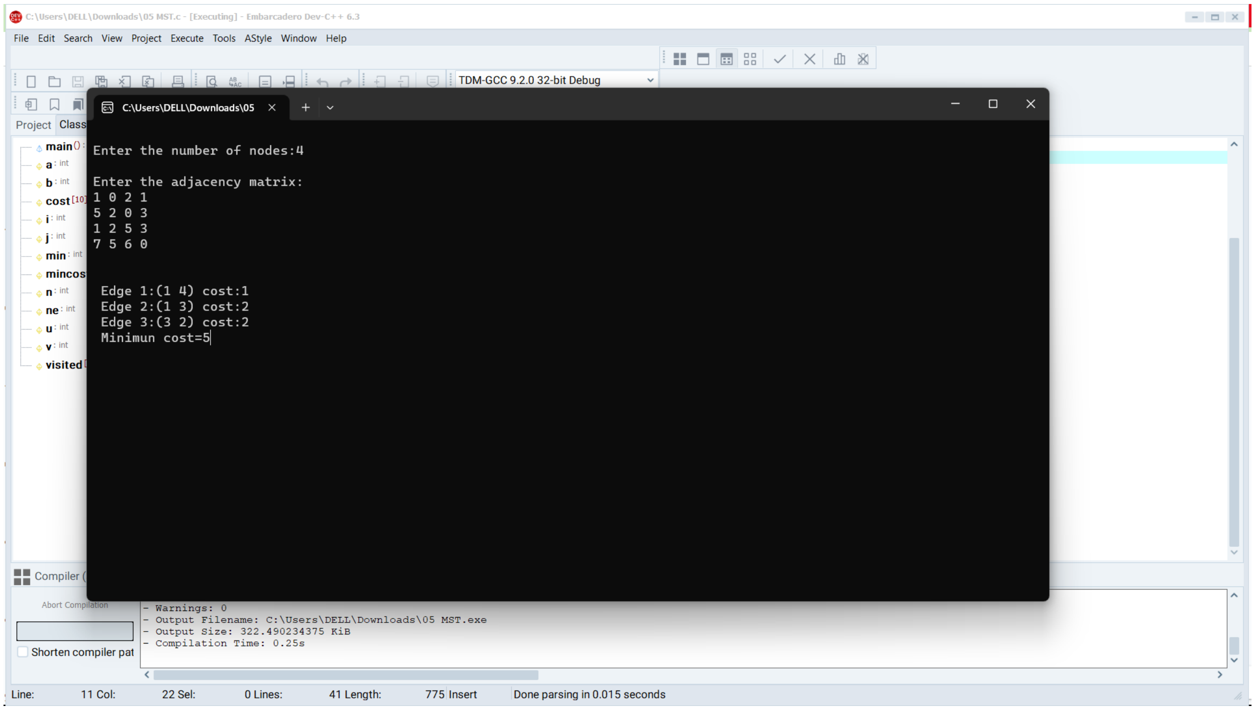
```
Enter the no. of objects:- 6

Enter the wts and profits of each object:-
2 33
5 22
6 44
4 32
6 52
4 66

Enter the capacityacity of knapsack:- 10

The result vector is:- 1.000000 1.000000        0.666667        0.000000        0.000000        0.000000
Maximum profit is:- 133.666672
--------------------------------
Process exited after 37.49 seconds with return value 0
Press any key to continue . . .
```

C:\Users\DELL\Downloads\04 Knapsack.c - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 32-bit Debug
Project Classes Debug
[*] Untitled1
01 Merge Sort.c
02 Divide and Conquerr.c
03 Prime Series.c
04 Knapsack.c
05 MST.c
knapsack (int n, float weight[],
main () : int
C:\Users\DELL\Downloads\04
Compiler (3)
Resource
Abort Compilation
Shorten compiler pat
Line: 40 Col: 31 Sel: 0 Lines: 59 Length: 1578 Insert Done parsing in 0.015 seconds

5. Write a program to perform MST using greedy techniques.

```
#include<stdio.h>
#include<conio.h>
int a,b,u,v,n,i,j,ne=1;
int visited[10]={0},min,mincost=0,cost[10][10];
void main()
{
        printf("\nEnter the number of nodes:");
        scanf("%d",&n);
        printf("\nEnter the adjacency matrix:\n");
        for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
        {
                scanf("%d",&cost[i][j]);
                if(cost[i][j]==0)
                        cost[i][j]=999;
        }
        visited[1]=1;
        printf("\n");
        while(ne < n)
        {
                for(i=1,min=999;i<=n;i++)
                for(j=1;j<=n;j++)
                if(cost[i][j]< min)
                if(visited[i]!=0)
                {
                        min=cost[i][j];
                        a=u=i;
                        b=v=j;
                }
                if(visited[u]==0 || visited[v]==0)
                {
                        printf("\n Edge %d:(%d %d) cost:%d",ne++,a,b,min);
                        mincost+=min;
                        visited[b]=1;
                }
                cost[a][b]=cost[b][a]=999;
        }
        printf("\n Minimun cost=%d",mincost);
        getch();
}
```

OUTPUT:



C:\Users\DELL\Downloads\05 MST.c - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 32-bit Debug
Project Class
main()
a : int
b : int
cost [10]
i : int
j : int
min : int
mincos
n : int
ne : int
u : int
v : int
visited
C:\Users\DELL\Downloads\05
Compiler (
Abort Compilation
Shorten compiler pat

```
Enter the number of nodes:4

Enter the adjacency matrix:
1 0 2 1
5 2 0 3
1 2 5 3
7 5 6 0


 Edge 1:(1 4) cost:1
 Edge 2:(1 3) cost:2
 Edge 3:(3 2) cost:2
 Minimun cost=5
```

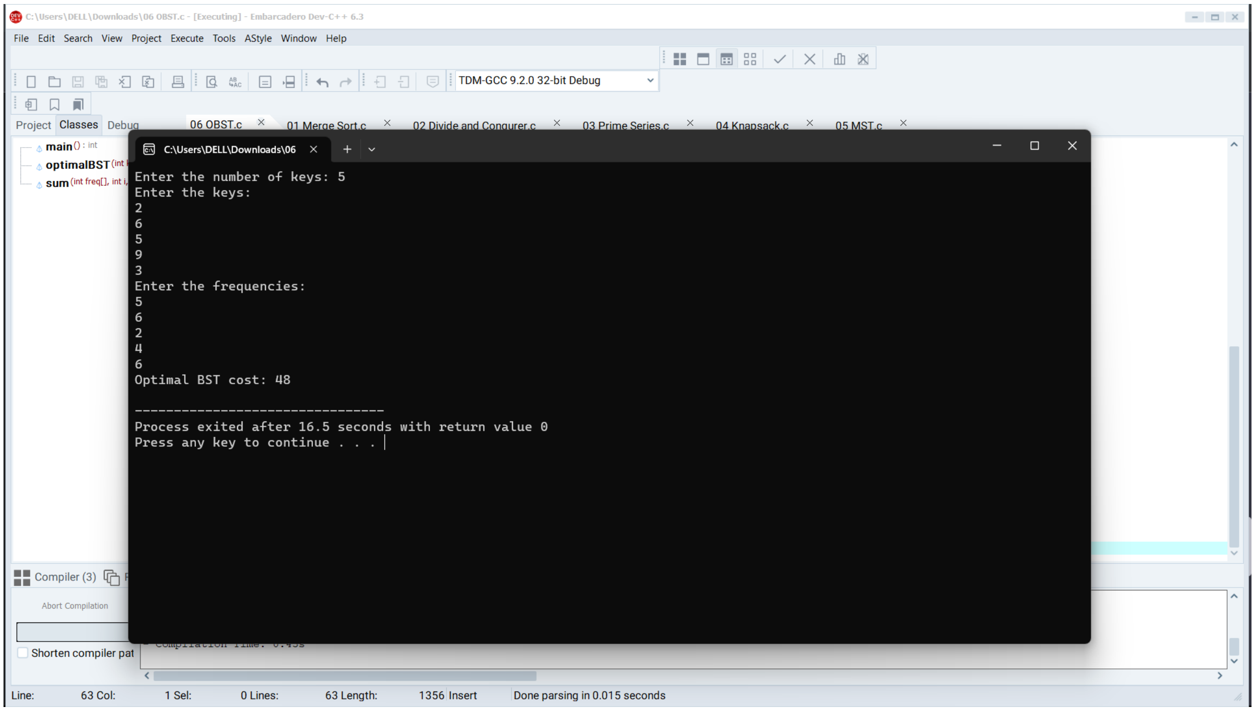
```
- Warnings: 0
- Output Filename: C:\Users\DELL\Downloads\05 MST.exe
- Output Size: 322.490234375 KiB
- Compilation Time: 0.25s
```

Line: 11 Col: 22 Sel: 0 Lines: 41 Length: 775 Insert Done parsing in 0.015 seconds

6. Using Dynamic programming concept to find out optimal binary search tree.

```
#include <stdio.h>
#include <limits.h>

#define MAX_KEYS 100

int sum(int freq[], int i, int j) {
    int s = 0;
    for (int k = i; k <= j; k++) {
        s += freq[k];
    }
    return s;
}

int optimalBST(int keys[], int freq[], int n) {
    int cost[n][n];

    for (int i = 0; i < n; i++) {
        cost[i][i] = freq[i];
    }
    for (int length = 2; length <= n; length++) {
        for (int i = 0; i <= n - length + 1; i++) {
            int j = i + length - 1;
            cost[i][j] = INT_MAX;

            for (int r = i; r <= j; r++) {
                int c = ((r > i) ? cost[i][r - 1] : 0) +
                        ((r < j) ? cost[r + 1][j] : 0) +
                        sum(freq, i, j);

                if (c < cost[i][j]) {
                    cost[i][j] = c;
                }
            }
        }
    }

    return cost[0][n - 1];
}

int main() {
    int n;
    printf("Enter the number of keys: ");
    scanf("%d", &n);

    int keys[MAX_KEYS];
    int freq[MAX_KEYS];

    printf("Enter the keys:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &keys[i]);
    }

    printf("Enter the frequencies:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &freq[i]);
    }

    printf("Optimal BST cost: %d\n", optimalBST(keys, freq, n));

    return 0;
}
```

OUTPUT:



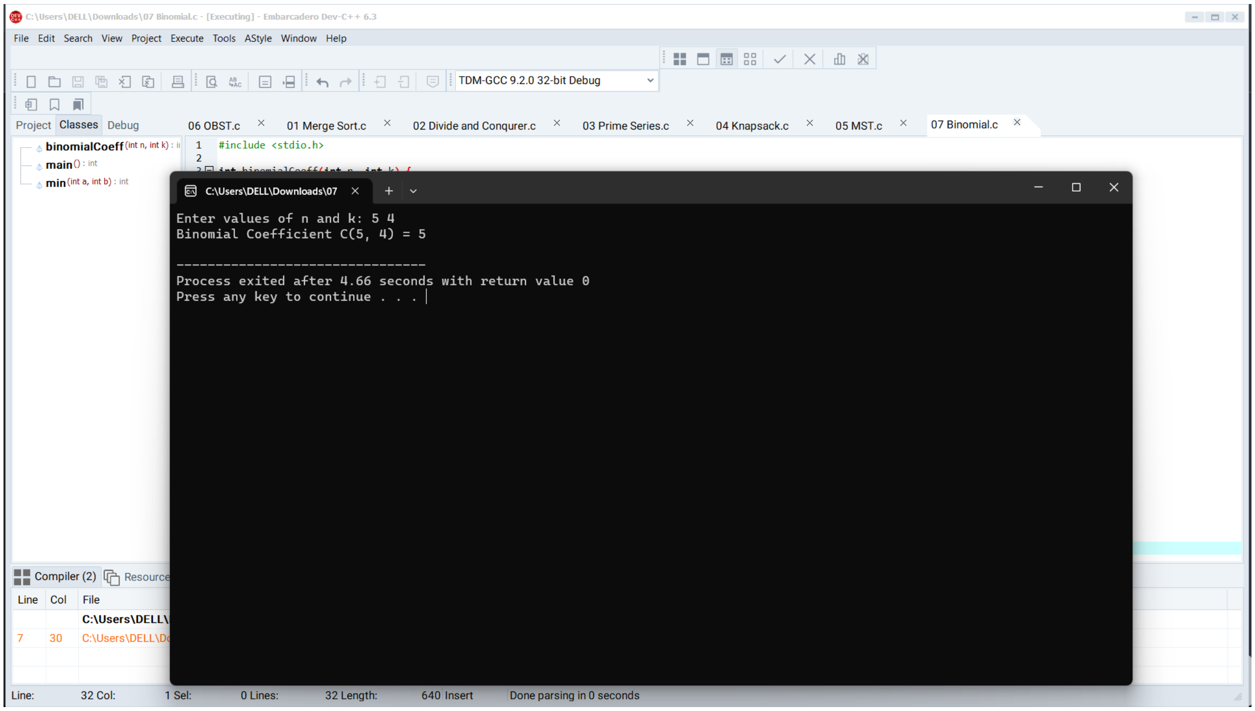
```
Enter the number of keys: 5
Enter the keys:
2
6
5
9
3
Enter the frequencies:
5
6
2
4
6
Optimal BST cost: 48

--------------------------------
Process exited after 16.5 seconds with return value 0
Press any key to continue . . .
```

Line: 63 Col: 1 Sel: 0 Lines: 63 Length: 1356 Insert Done parsing in 0.015 seconds

7. Using Dynamic programming techniques to find binomial coefficient of a given number

```
#include <stdio.h>

int binomialCoeff(int n, int k) {
    int C[n + 1][k + 1];

    for (int i = 0; i <= n; i++) {
        for (int j = 0; j <= min(i, k); j++) {
            if (j == 0 || j == i)
                C[i][j] = 1;
            else
                C[i][j] = C[i - 1][j - 1] + C[i - 1][j];
        }
    }

    return C[n][k];
}

int min(int a, int b) {
    return (a < b) ? a : b;
}

int main() {
    int n, k;

    printf("Enter values of n and k: ");
    scanf("%d %d", &n, &k);

    printf("Binomial Coefficient C(%d, %d) = %d\n", n, k, binomialCoeff(n, k));

    return 0;
}
```

OUTPUT:



C:\Users\DELL\Downloads\07 Binomial.c - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 32-bit Debug
Project Classes Debug
06 OBST.c
01 Merge Sort.c
02 Divide and Conquer.c
03 Prime Series.c
04 Knapsack.c
05 MST.c
07 Binomial.c
binomialCoeff (int n, int k)
main () : int
min (int a, int b) : int
C:\Users\DELL\Downloads\07
Compiler (2)
Resources
Line Col File
Line: 32 Col: 1 Sel: 0 Lines: 32 Length: 640 Insert Done parsing in 0 seconds

```
#include <stdio.h>
```

```
Enter values of n and k: 5 4
Binomial Coefficient C(5, 4) = 5

--------------------------------
Process exited after 4.66 seconds with return value 0
Press any key to continue . . .
```