SR ENGINEERING COLLEGE

Approved by AICTE, New Delhi | Affiliated to JNTUH | Accredited by NBA

Ananthasagar (V), Hasanparthy (M), WARANGAL - 506 371

Autonomous

PROBLEM SOLVING WITH PROGRAMMING

(PSP)

Project by - Karnakanti Raviteja

Under the Guidance of

Mr. P Pramod Kumar

Senior Assistant Professor

PROBLEM STATEMENT:

Develop a C Application to store details of 'n 'Trains in terms TrainName, TrainType, TrainNumber, TrainOrigin and TrainDestination. Allocate a memory using Dynamic

Memory Management functions.

Provide the functionality for below mentioned:

1. Read ‘n’ trains details dynamically.
2. Sort (Ascending /Descending) ‘n’ trains details:
    1. Name Wise
    2. Type Wise
    3. Number Wise
    4. Origin Wise
    5. Destination Wise
3. Search ‘n’ trains details:
    1. Name Wise
    2. Type Wise
    3. Number Wise
    4. Origin Wise
    5. Destination Wise
4. Print ‘n’ trains details.

MODULES:

Main:

We declared all variables and Structures as Global Variables.

We can choose any function with help their function calls placed in switch case. In order to repeat this we have used looping statement(while) with a condition.

We used four modules for this project.

1. Read / Input:

In this module the program will asks you to enter details of ‘n’ trains .

To give n trains details we used lopping constructs (for loop).

2. Sorting:

In this module we have a Sub-menu. We used control statement(switch case), so that we can sort namewise , typewise, numberwise, origin wise, destinationwise easily in ascending order. If we press 1 the program will sort namewise, if we press 2 the program will sort typewise, if we press 3 the program will sort number wise, if we press 4 the program will sort origin wise, if we press 5 the program will sort destinationwise.

We used another control statement(if) so that it will ask you whether you want to continue sort operation again or you want to continue with other modules like search, print etc.

3. Searching:

In this module we have a Sub-menu. We used control statements (switch case), so that we can search name wise , type wise, number wise, origin wise, destination wise easily. If we press 1 the program will search name wise, if we press 2 the program will search type wise, if we press 3 the program will search number wise, if we press 4 the program will search origin wise, if we press 5 the program will search destination wise.

We used another control statement (if) so that it will ask you whether you want to continue search operation again or you want to continue with other modules like sort, print etc.

4. Print:

In this module entire ‘n’ trains details will be displayed with help of printf and looping constructs (for loop).

KNOWLEDGE REQUIRED TO DEVELOP THIS APPLICATION:

->Control Statements (if, if-else, switch)

->looping Statements (for, while)

->Arrays (1-d array)

->Strings (strings, table of strings)and its functions(strcpy, strcmp)

->Functions (any category of UDF functions)

->Structures (structure/ nested structure)

->Pointers (pointers to strings, pointers to structures)

->Dynamic Memory Allocation (malloc() / calloc() and free()).

SOURCE CODE:

```
//Team project on Train(s) details interms of Train name,Train type,Trainnumber,Train origin,Train destination.

// header files

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

struct train                    /*Structure defination*/

        {

                char tn[20];

                char tt[20];

                int tnum;

                char to[20];

                char td[20];

        };

void read(void);                /*functions prototype*/

void sort(void);

void search(void);

void print(void);

void Namewise();

void Typewise();
```

```
void Numberwise();
void Orginwise();
void Destinationwise();
void Namewise1();
void Typewise1();
void Numberwise1();
void Orginwise1();
void Destinationwise1();
                                //global variables
char a,tname[10],ttype[10],torigin[10],tdes[10],str[20][50],temp[200];
int ch,i,tnumber,j,c=0,tns,y;
struct train *t,*ptr;                   //pointer to a structure
                                //main starts here
int main()
{
        printf("\n Enter the number of trains :");//reading number of trains dynamically
        scanf("%d",&tns);
        t=(struct train *) malloc(tns*sizeof(struct train));// Dynamic memory allocation
   ptr=t;
                                        // MAIN-MENU repetation
        do
```

```
        {
        printf("\n***MAIN-MENU:*** \n Press '1' to Read input \n Press '2' to Sort trains
\n Press '3' to Search for the train \n Press '4' to PrintDetails \n");
        printf("\n Enter option:");
        scanf("%d",&ch);
                                //switch case
        switch(ch)
        {
                case 1: read();//function call
                break;
                case 2: sort();//function call
                break;
                case 3:search();//function call
                break;
                case 4:print();//function call
                break;
                default:printf("\n Sorry you entered wrong option , try again");
        }
        printf("\n Enter y or Y to repeat MAIN-MENU \n Enter n or N  to exit from MAIN-
MENU:");
        scanf(" %c",&a);
    }
```

```
        while(a=='y'||a=='Y');
        return 0;
}
//Function definitions
void read(void)
        {
        printf("\n Enter details of train's':");
        for(i=0;i<tns;i++)
        {
                printf("\n Enter details of train' %d ':",i+1);
                printf("\n Enter TrainName:");
                scanf("%s",t->tn);
                printf("\n Enter TrainType:");
                scanf("%s",t->tt);
                printf("\n Enter TrainNumber:");
                scanf("%d",&t->tnum);
                printf("\n Enter TrainOrgin:");
                scanf("%s",t->to);
                printf("\n Enter TrainDestination:");
                scanf("%s",t->td);
                t++;
```

```
        }
        }
void sort()
{

        // SUB-MENU repetation
        do
        {
        printf("\n SUB-MENU FOR SORTING: \n press 1. to sort NAMEWISE \n press 2. to sort TYPEWISE \n press 3. to sort Numberwise \n press 4.to sort Orginwise \n press 5. to sort Destinationwise\n");
        scanf("%d",&ch);
                switch(ch) //switch case
        {
                case 1: Namewise();//function call
                break;
                case 2: Typewise();//function call
                break;
                case 3:Numberwise();//function call
                break;
                case 4:Orginwise();//function call
                break;
```

```
        case 5:Destinationwise();//function call
                    break;
                    default:printf("\n sorry you entered wrong option , try again");
            }
                    printf("\n enter y or Y to continue sort operation again \n enter n or N  to
go to main menu");
            scanf(" %c",&a);
            if(a=='n'||a=='N')
            return;
       }
            while(a=='y'||a=='Y');
            }
void search()
{
            // SUB-MENU repetation
            do
            {
            printf("\n  SUB-MENU FOR SEARCHING: \n 1. NAMEWISE \n 2. TYPEWISE \n 3.
Numberwise \n 4. Orginwise \n. 5. Destinationwise\n");
            scanf("%d",&ch);
            //switch case
            switch(ch)
```

```
        {
                case 1: Namewise1();//function call
                break;
                case 2: Typewise1();//function call
                break;
                case 3:Numberwise1();//function call
                break;
                case 4:Orginwise1();//function call
                break;
      case 5:Destinationwise1();//function call
                break;
                default:printf("\n sorry you entered wrong option , try again");
        }
                printf("\n enter y or Y to continue search operation again ,enter n or N  to
go to main menu");
        scanf(" %c",&a);
        if(a=='n'||a=='N')
        return;
    }   while(a=='y'||a=='Y');
        }
        // FunctSion definations to sort train details
```

```
        // function defination to sort train names
        void Namewise()
{
        t=ptr;
        for(i=0;i<tns;i++)
        {
        strcpy(str[i],t->tn);
        t++;
}
         for(i=0;i<tns;i++)
         {
         for(j=i+1;j<tns;j++)
         {
         y=strcmp(str[i],str[j]);
         if(y>0)
         {
          strcpy(temp,str[i]);
          strcpy(str[i],str[j]);
          strcpy(str[j],temp);
}
}
```

```
}
            printf("\n Sorted trainnames :");
             for(i=0;i<tns;i++)
                    puts(str[i]);
          }
          // function defination to sort train types
          void Typewise()
{
          t=ptr;
          for(i=0;i<tns;i++)
          {
           strcpy(str[i],t->tt);
           t++;
    }
           for(i=0;i<tns;i++)
           {
           for(j=i+1;j<tns;j++)
           {
           y=strcmp(str[i],str[j]);
           if(y>0)
           {
```

```
            strcpy(temp,str[i]);
            strcpy(str[i],str[j]);
            strcpy(str[j],temp);
}
}

            printf("\n Sorted traintypes:");
             for(i=0;i<tns;i++)
                    puts(str[i]);
          }
}
          // function defination to sort train numbers
          void Numberwise()
{
          t=ptr;
                    int tempnum, number[i];
                    for(i=0;i<tns;i++)
                    {
            number[i]=t->tnum;
          t++;
}
```

```
        for(i=0;i<tns;i++)
        {
                for(j=i+1;j<tns;j++)
                {
                        if(number[i]>number[j])
                        {
                                tempnum=number[i];
                                number[i]=number[j];
                                number[j]=tempnum;
                        }
                }
        }
        printf("\n sorted train numbers:");
        for(i=0;i<tns;i++)
        {
                printf("\n %d",number[i]);
        }
}
// Function defination to sort trains origin
        void Orginwise()
{
```

```
        t=ptr;
        for(i=0;i<tns;i++)
        {
        strcpy(str[i],t->to);
         t++;
}


 for(i=0;i<tns;i++)
        {
        for(j=i+1;j<tns;j++)
        {
        y=strcmp(str[i],str[j]);
        if(y>0)
        {
         strcpy(temp,str[i]);
         strcpy(str[i],str[j]);
         strcpy(str[j],temp);
}
}
}
```

```
            printf("\n Sorted train origins:");
        for(i=0;i<tns;i++)
            puts(str[i]);
    }


// function defination to sort trains destinations
    void Destinationwise()
{
    t=ptr;
    for(i=0;i<tns;i++)
    {
    strcpy(str[i],t->td);
     t++;
}
     for(i=0;i<tns;i++)
     {
     for(j=i+1;j<tns;j++)
     {
     y=strcmp(str[i],str[j]);
     if(y>0)
```

```
        {
         strcpy(temp,str[i]);
         strcpy(str[i],str[j]);
         strcpy(str[j],temp);
}
}
}
          printf("\n Sorted train destinations:");
           for(i=0;i<tns;i++)
                puts(str[i]);
        }
        //function defination to search trains details
        // function defination to search train names
        void Namewise1()
{
        t=ptr;
        printf("\n Enter train name:");
        scanf("%s",tname);
                for(i=0;i<tns;i++)
        {
                char x=strcmp(tname,t->tn);
```

```
                if(x==0)
                {
                        printf("\n The details of train %d like name,type,number,origin,destination:\n",i+1);
                printf("Name: %s\n",t->tn);
                printf("Type: %s\n",t->tt);
                printf("Number: %d\n",t->tnum);
                printf("Orgin: %s\n",t->to);
                printf("Destination: %s\n",t->td);
                c++;
                }
                t++;
        }
                if(c==0)
                        {
                        printf("\n sorry you entered name is not their please try again");
                }
        }
        // function defination to search train names
        void Typewise1()
{
```

```
                t=ptr;
        printf("\n Enter train type:");
        scanf("%s",ttype);
        for(i=0;i<tns;i++)
        {
                char x=strcmp(ttype,t->tt);
                if(x==0)
                {
                        printf("\n The details of train %d like
name,type,number,origin,destination:\n",i+1);
                printf("Name: %s\n",t->tn);
                printf("Type: %s\n",t->tt);
                printf("Number: %d\n",t->tnum);
                printf("Orgin: %s\n",t->to);
                printf("Destination: %s\n",t->td);
      c++;
                }
                t++;
        }
        if(c==0)
                {
```

```
                printf("\n sorry you entered train type is not their please try again");
            }


        }
        // function defination to search train number
        void Numberwise1()
{
        t=ptr;
        printf("\n Enter train number:");
        scanf("%d",&tnumber);
    for(i=0;i<tns;i++)
        {
            if(tnumber==t->tnum)
            {
                printf("\n The details of train %d like name,type,number,origin,destination:\n",i+1);
            printf("Name: %s\n",t->tn);
            printf("Type: %s\n",t->tt);
            printf("Number: %d\n",t->tnum);
            printf("Orgin: %s\n",t->to);
            printf("Destination: %s\n",t->td);
```

```
            c++;
                }
                t++;
        }
        if(c==0)
                {
                        printf("\n sorry you entered train number is not their please try
again");
                }
        }
        // function defination to search trains origin
        void Orginwise1()
{
        t=ptr;
        printf("\n  Enter origin of train:");
        scanf("%s",torigin);
        for(i=0;i<tns;i++)
        {
                char x=strcmp(torigin,t->to);
                if(x==0)
                {
```

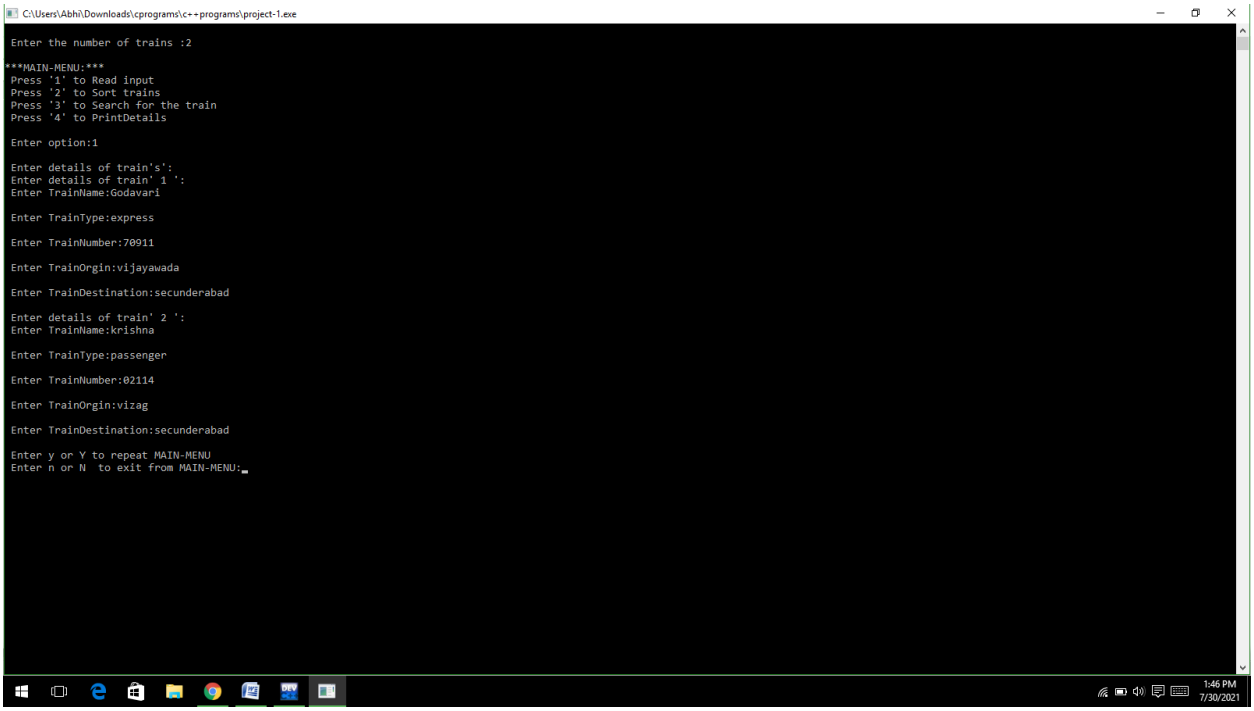
```
                printf("\n The details of train %d like name,type,number,origin,destination:\n",i+1);
                printf("Name: %s\n",t->tn);
                printf("Type: %s\n",t->tt);
                printf("Number: %d\n",t->tnum);
                printf("Orgin: %s\n",t->to);
                printf("Destination: %s\n",t->td);
                c++;
                }
                t++;
        }
        if(c==0)
                {
                        printf("\n sorry you entered train origin is not their please try again");
                }
}

// function defination to search train destinations
        void Destinationwise1()
{
        t=ptr;
```

```
        printf("\n Enter destination of train:");
        scanf("%s",tdes);
        for(i=0;i<tns;i++)
        {
                char x=strcmp(tdes,t->td);
                if(x==0)
                {
                   printf("\n The details of train %d like name,type,number,origin,destination:\n",i+1);
                printf("Name: %s\n",t->tn);
                printf("Type: %s\n",t->tt);
                printf("Number: %d\n",t->tnum);
                printf("Orgin: %s\n",t->to);
                printf("Destination: %s\n",t->td);
            c++;
                }
                t++;
        }
                if(c==0)
                {
                        printf("\n sorry you entered train destination is not their please try again");
```

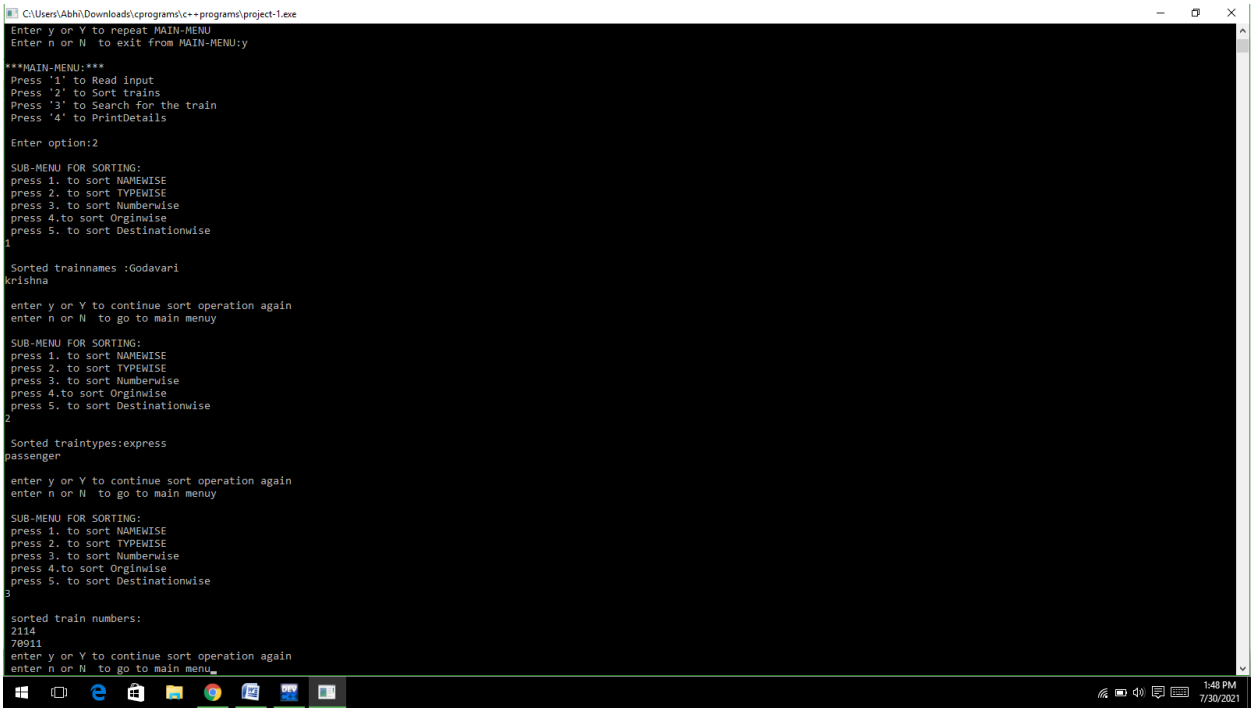
```
            }
        }
        // function defination to print trains details
void print()
{

printf("\n The details of trains like trainname,type,number,origin,destination:");
        t=ptr;
        for(i=0;i<tns;i++)
        {
            printf("\n The details of train %d like name,type,number,origin,destination:\n",i+1);
            printf("Name: %s\n",t->tn);
            printf("Type: %s\n",t->tt);
            printf("Number: %d\n",t->tnum);
            printf("Orgin: %s\n",t->to);
            printf("Destination: %s\n",t->td);
            t++;
   }
}
```

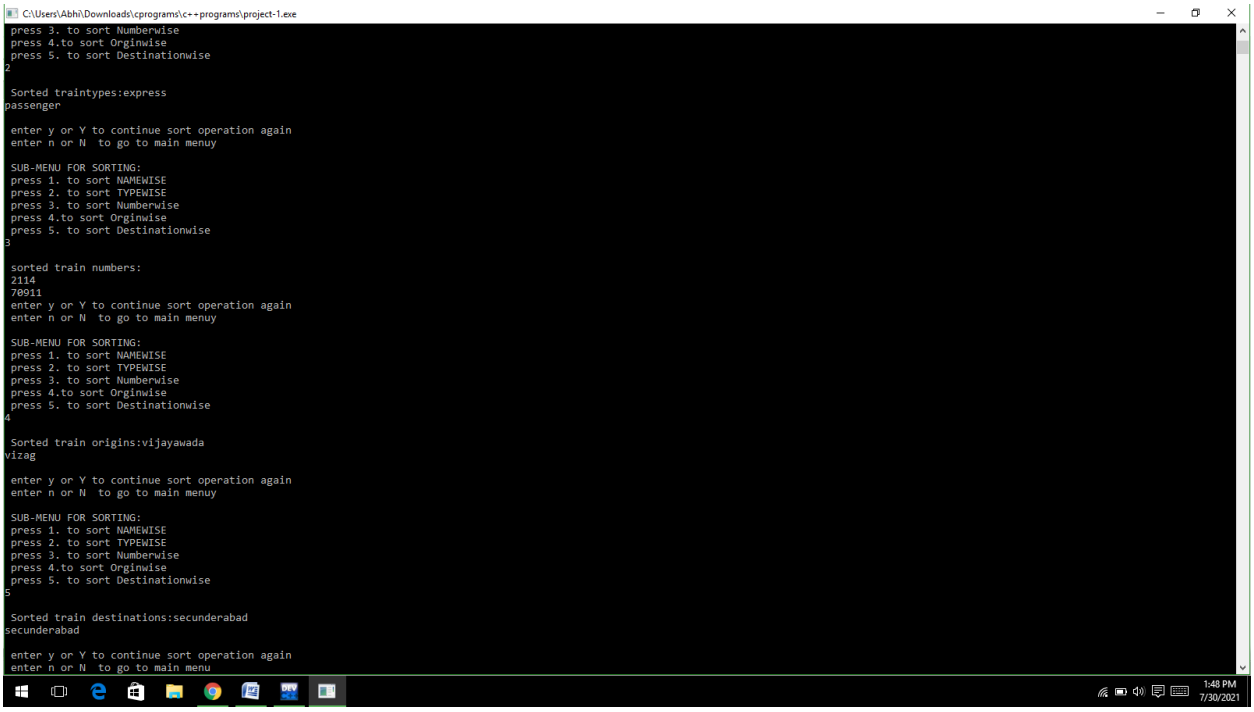
RESULT:

Readinginput:

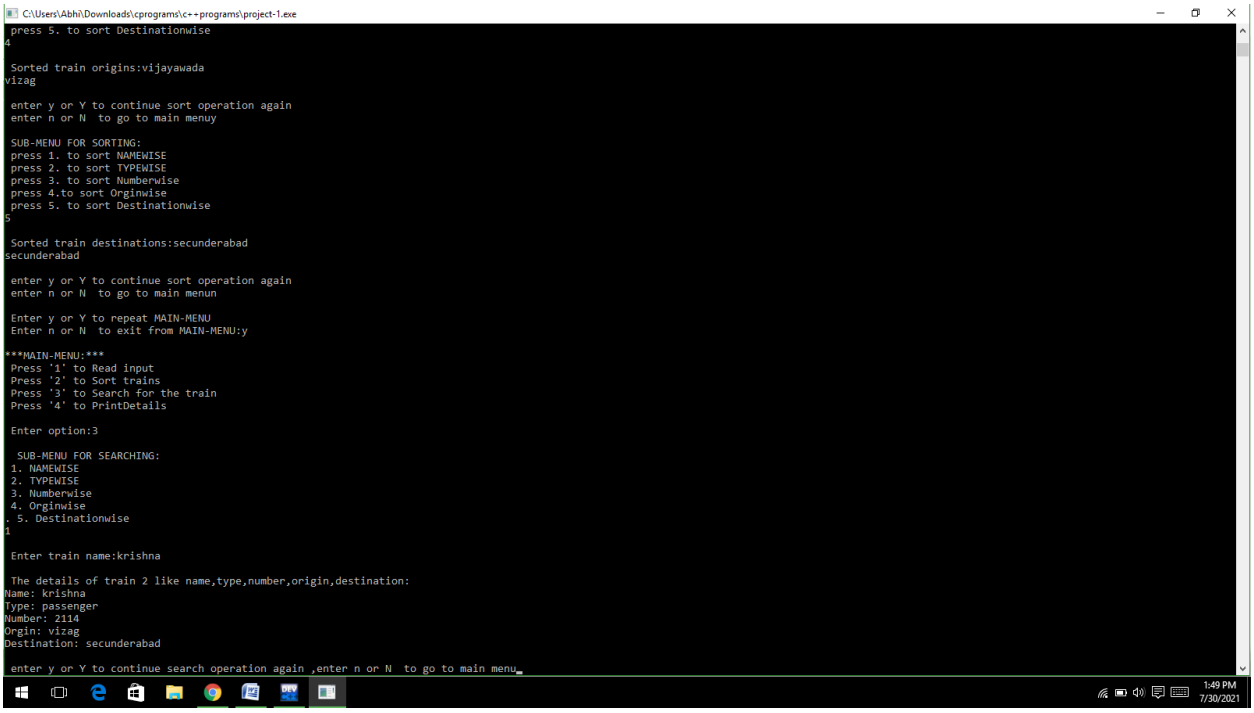


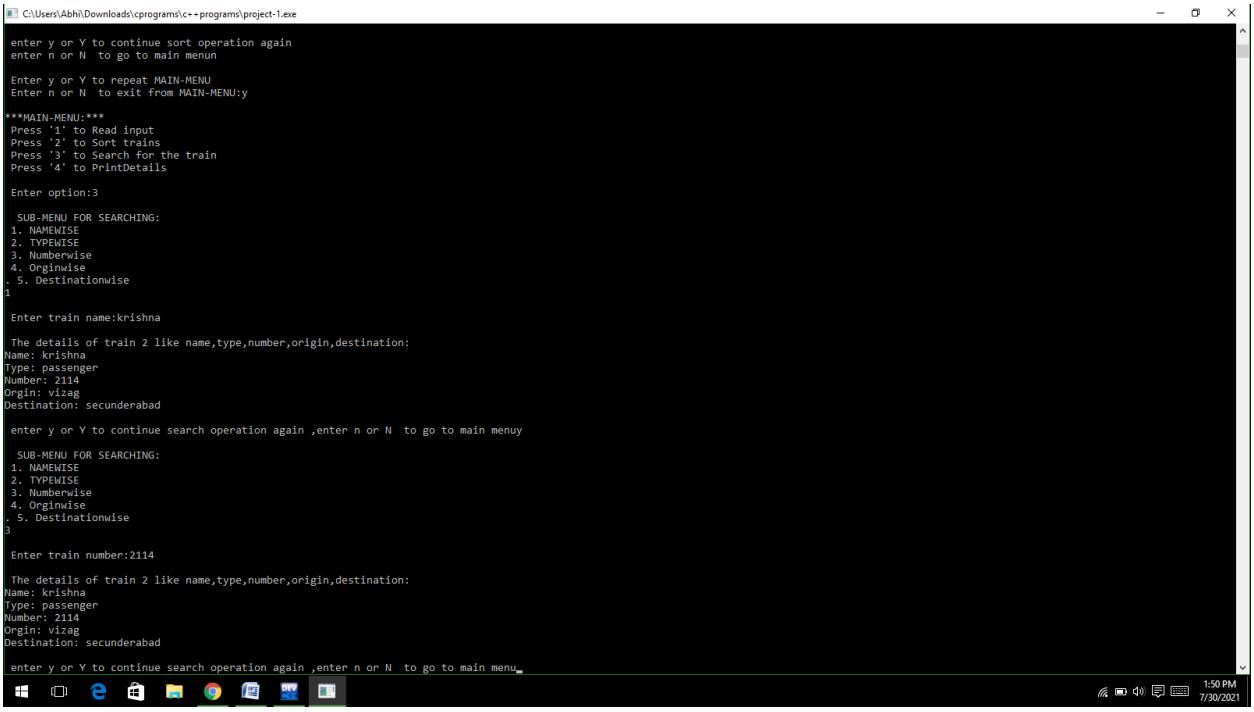
Sorting trains:



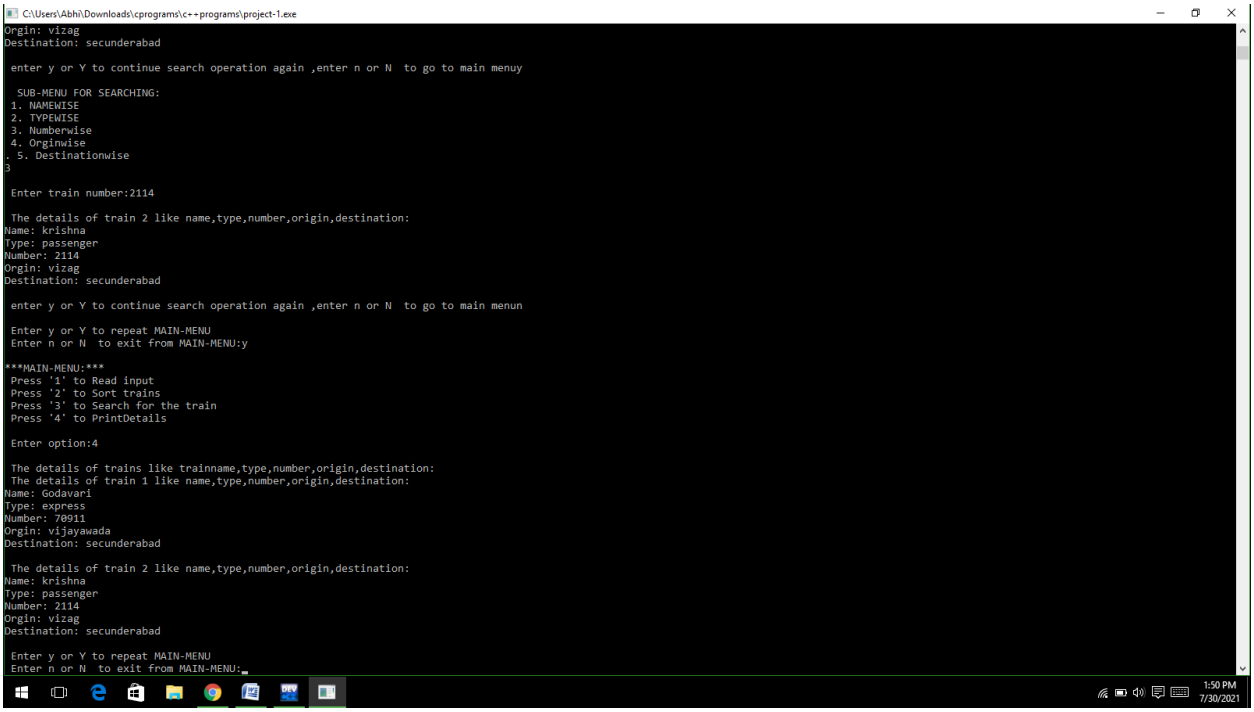


Searching trains:





Printing details:


C:\Users\Abhi\Downloads\cprograms\c++programs\project-1.exe
Orgin: vizag
Destination: secunderabad
enter y or Y to continue search operation again ,enter n or N to go to main menuy
SUB-MENU FOR SEARCHING:
1. NAMEWISE
2. TYPEWISE
3. Numberwise
4. Orginwise
. 5. Destinationwise
3
Enter train number:2114
The details of train 2 like name,type,number,origin,destination:
Name: krishna
Type: passenger
Number: 2114
Orgin: vizag
Destination: secunderabad
enter y or Y to continue search operation again ,enter n or N to go to main menun
Enter y or Y to repeat MAIN-MENU
Enter n or N to exit from MAIN-MENU:y
MAIN-MENU:
Press '1' to Read input
Press '2' to Sort trains
Press '3' to Search for the train
Press '4' to PrintDetails
Enter option:4
The details of trains like trainname,type,number,origin,destination:
The details of train 1 like name,type,number,origin,destination:
Name: Godavari
Type: express
Number: 70911
Orgin: vijayawada
Destination: secunderabad
The details of train 2 like name,type,number,origin,destination:
Name: krishna
Type: passenger
Number: 2114
Orgin: vizag
Destination: secunderabad
Enter y or Y to repeat MAIN-MENU
Enter n or N to exit from MAIN-MENU:
1:50 PM
7/30/2021