Practical File

Data Structures Lab – 20CP201P

Subject: Data Structures Lab – 20CP201P

Branch: B.Tech Computer Engineering

Division: 3

Group: G5

Project Team: 1

Team Members:

REHA SHAH – 21BCP148 (Group Leader)

VRUNDA BATHIA – 21BCP168

TANYA KHUNTETA – 21BCP163

MISARI BALDHA – 21BCP173

KRUPA PATEL – 21BCP144

INDEX

Project 1: Structures

118 ELEMENTS PROPERTIES

Code:

```
//task - create a structure with real-life applications
//PERIODIC TABLE APPLICATION

#include<stdio.h>
#include<math.h>
#include<string.h>

//structure for periodic table elements information
struct ptable
{
   char symbol[4]; //element symbol
   char name[20]; //element name
   int ano; //atomic number
   float am; //atomic mass
   char type[40]; //type of element
   char block; //block of periodic table
   int orbits; //number of orbits in ground state on atom
   char state; //state of element at room temperature
};

//main code
void main()
{
   printf("LET'S LEARN ABOUT THE PERIODIC TABLE !!\n");
   //structure object initiation for all 118 elements
```

```
struct ptable elements[118] = {{.symbol="H" ,.name="Hydrogen" , .ano=1 ,.am=1.00 ,.type="Non-Metal" , .block='s' ,.orbits=1 ,.state='g' },

{.symbol="He",.name="Helium" ,   .ano=2 ,.am=4.00 ,.type="Noble Gas" , .block='p' ,.orbits=1 ,.state='g' },

{.symbol="Li",.name="Lithium" ,  .ano=3 ,.am=7.00,.type="Alkali Metal" ,.block='s' ,.orbits=2 ,.state='s' },

{.symbol="Be",.name="Beryllium" ,.ano=4 ,.am=9.01 ,.type="Alkaline Earth Metal" ,.block='s',.orbits=2 ,.state='s' },

{.symbol="B",.name="Boron" , .ano=5 ,.am=10.81 ,.type="Metalloid" ,  .block='p' ,.orbits=2 ,.state='s' },

{.symbol="C",.name="Carbon" , .ano=6 ,.am=12.01 ,.type="Non-Metal" ,  .block='p' ,.orbits=2 ,.state='s' },

{.symbol="N",.name="Nitrogen" , .ano=7 ,.am=14.00 ,.type="Non-Metal" ,  .block='p' ,.orbits=2 ,.state='g' },

{.symbol="O",.name="Oxygen" , .ano=8 ,.am=15.99 ,.type="Non-Metal" ,  .block='p' ,.orbits=2 ,.state='g'},

{.symbol="F",.name="Fluorine" , .ano=9 ,.am=18.99 ,.type="Halogen" , .block='p',.orbits=2 ,.state='g' },

{.symbol="Ne",.name="Neon" , .ano=10 ,.am=20.17 ,.type="Noble Gas" ,.block='p' ,.orbits=2,.state='g' },

{.symbol="Na" ,.name="Sodium" , .ano=11 ,.am=22.98 ,.type="Alkali Metal" ,.block='s' ,.orbits=3 ,.state='s' },

{.symbol="Mg" ,.name="Magnesium" , .ano=12 ,.am=24.30 ,.type="Alkaline Earth Metal" ,.block='s',.orbits=3 ,.state='s' },

{.symbol="Al" ,.name="Aluminum" , .ano=13 ,.am=26.98 ,.type="Post Transition Metal" ,.block='p' ,.orbits=3 ,.state='s'},

{.symbol="Si" ,.name="Silicon" , .ano=14 ,.am=28.08 ,.type="Metalloid" , .block='p' ,.orbits=3 ,.state='s' },

{.symbol="P" ,.name="Phosphorus" , .ano=15 ,.am=30.97 ,.type="Non-Metal" , .block='p' ,.orbits=3 ,.state='s' },

{.symbol="S" ,.name="Sulfur" , .ano=16 ,.am=32.06 ,.type="Non-Metal" , .block='p' ,.orbits=3 ,.state='s' },

{.symbol="Cl" ,.name="Chlorine" , .ano=17 ,.am=35.45 ,.type="Halogen" ,  .block='p' ,.orbits=3 ,.state='g'},

{.symbol="Ar" ,.name="Argon" , .ano=18 ,.am=39.94 ,.type="Noble Gas" ,.block='p' ,.orbits=3 ,.state='g' },
```

```
{.symbol="K" ,.name="Potassium" , .ano=19 ,.am=39.09 ,.type="Alkali Metal" ,.block='s' ,.orbits=4 ,.state='s' },

{.symbol="Ca" ,.name="Calcium" , .ano=20 ,.am=40.07 ,.type="Alkaline Earth Metal" ,.block='s' ,.orbits=4 ,.state='s'},

{.symbol="Sc" ,.name="Scandium" , .ano=21 ,.am=44.95 ,.type="Transition Metal" ,.block='d',.orbits=4 ,.state='s' },

{.symbol="Ti" ,.name="Titanium" , .ano=22 ,.am=47.86 ,.type="Transition Metal" ,.block='d',.orbits=4 ,.state='s' },

{.symbol="V" ,.name="Vanadium" , .ano=23 ,.am=50.94,.type="Transition Metal" , .block='d' ,.orbits=4 ,.state='s'},

{.symbol="Cr" ,.name="Chromium" , .ano=24 ,.am=51.99 ,.type="Transition Metal" ,.block='d' ,.orbits=4 ,.state='s' },

{.symbol = "Mn" ,.name = "Manganese" ,.ano = 25 ,.am =54 ,.type ="Transition Metal" ,.block ='d' ,.orbits =4 ,.state ='solid' },

{.symbol = "Fe" ,.name = "Iron" ,.ano = 26,.am =55 ,.type ="Transition Metal" ,.block ='d' ,.orbits =4 ,.state ='solid' },

{.symbol = "Co" ,.name = "Cobalt" ,.ano = 27 ,.am =58 ,.type ="Transition Metal" ,.block ='d' ,.orbits =4 ,.state ='solid' },

{.symbol = "Ni" ,.name = "Nickel" ,.ano =28 , .am =58 ,.type ="Transition Metal" ,.block ='d' ,.orbits =4 ,.state ='solid' },

{.symbol = "Cu" ,.name = "Copper" ,.ano =29 , .am =63 ,.type ="Transition Metal" ,.block ='d' ,.orbits =4 ,.state ='solid' },

{.symbol = "Zn",.name = "Zinc" ,.ano =30 , .am =65 ,.type ="Transition Metal" ,.block ='d' ,.orbits =4 ,.state = 'solid' },

{.symbol = "Ga" ,.name = "Gallium" ,.ano =31 ,.am =69 ,.type ="Post-Transition Metal" ,.block ='d' ,.orbits =4 ,.state ='solid' },

{.symbol = "Ge" ,.name = "Germanium" ,.ano =32 ,.am =72 ,.type ="Mettaloid" ,.block ='p' ,.orbits =4 ,.state = 'solid' },

{.symbol = "As" ,.name = "Arsenic" ,.ano =33 ,.am =74 ,.type ="Mettaloid" ,.block ='p' ,.orbits =4 ,.state = 'solid' },

{.symbol = "Se" ,.name = "Selenium" ,.ano =34 ,.am =79 ,.type ="Non-Metal" ,.block ='p' ,.orbits =4 ,.state ='solid' },

{.symbol = "Br" ,.name = "Bromine" ,.ano =35 ,.am =80 ,.type ="Halogen" ,.block ='p' ,.orbits =4 ,.state ='liquid' },

{.symbol = "Kr" ,.name = "Krypton" ,.ano =36 ,.am =83 ,.type ="Noble Gas" ,.block ='p' ,.orbits =4 ,.state ='gas' },
```

```
{.symbol = "Rb" ,.name = "Rubidium" ,.ano =37 ,.am =85 ,.type ="Alkali Metal" ,.block ='s' ,.orbits =5 ,.state ='solid'},

{.symbol = "Sr" ,.name = "Stronium" ,.ano =38 ,.am =87 ,.type ="Alkaline Earth Metal" ,.block ='s' ,.orbits =5 ,.state ='solid'},

{.symbol = "Y" ,.name =  "Yttrium" ,.ano =39 ,.am =39 ,.type ="Transition Metal" ,.block ='d' ,.orbits =5 ,.state ='solid'},

{.symbol = "Zr" ,.name = "Zirconium" ,.ano =40 ,.am =91 ,.type ="Transition Metal" ,.block ='d' ,.orbits =5 ,.state ='solid'},

{.symbol = "Nb" ,.name = "Nibodium ",.ano =41 ,.am =92 ,.type ="Transition Metal" ,.block ='d' ,.orbits =5 ,.state ='solid'},

{.symbol = "Mo" ,.name = "Molybdenum" ,.ano =42 ,.am =96 ,.type ="Transition Metal" ,.block ='d' ,.orbits =5 ,.state ='solid'},

{.symbol = "Tc" ,.name = "Technetium",.ano =43 ,.am =98 ,.type ="Transition Metal" ,.block ='d' ,.orbits =5 ,.state ='solid'},

{.symbol = "Ru" ,.name = "Ruthenium" ,.ano =44 ,.am =101 ,.type ="Transition Metal" ,.block ='d' ,.orbits =5 ,.state ='solid' },

{.symbol = "Rh" ,.name = "Rhodium" ,.ano =45 ,.am =102 ,.type ="Transition Metal" ,.block ='d' ,.orbits =5 ,.state ='solid' },

{.symbol = "Pd" ,.name = "Palladium" ,.ano =46 ,.am =106 ,.type ="Transition Metal" ,.block ='d' ,.orbits =5 ,.state ='solid' },

{.symbol = "Ag" ,.name = "Silver" ,.ano =47 ,.am =107 ,.type ="Transition Metal" ,.block ='d' ,.orbits =5 ,.state ='solid' },

{.symbol = "Cd" ,.name = "Cadmium" ,.ano =48 ,.am =112 ,.type ="Transition Metal" ,.block ='d' ,.orbits =5 ,.state ='solid' },

{.symbol = "Cd", .name = "Cadmium", .ano =48 , .am =112.41, .type = "Post-Transition Metal", .block = 'd', .orbits = 5, .state = 's'},

{.symbol = "In", .name = "Indium",  .ano =49 , .am =114.81 , .type = "Post-Transition Metal", .block = 'p', .orbits = 5, .state = 's'},

{.symbol = "Sn", .name = "Tin", .ano =50 , .am =118.71 , .type = "Post-Transition Metal", .block = 'p', .orbits = 5, .state = 's'},

{.symbol = "Sb", .name = "Antimony", .ano = 52, .am =121.76 , .type = "Metalloid", .block = 'p', .orbits = 5, .state = 's'},

{.symbol = "Te", .name = "Tellurium", .ano = 53, .am =127.60 , .type = "Metalloid", .block = 'p', .orbits = 5, .state = 's'},

{.symbol = "I", .name = "Iodine", .ano = 54, .am =126.90 , .type = "Halogen", .block = 'p', .orbits = 5, .state = 's'},
```

```
    {.symbol = "Xe", .name = "Xenon", .ano = 55 , .am =131.29 , .type = "Noble Gas", .block = 'p', .orbits = 5, .state = 's'},

    {.symbol = "Cs", .name = "Caesium", .ano = 56 , .am =132.90 , .type = "Alkali Metal", .block = 's', .orbits = 6, .state = 's'},

    {.symbol = "Ba", .name = "Barium", .ano = 57, .am =137.32 , .type = "Alkaline Earth Metal", .block = 's', .orbits = 6, .state = 's'},

    {.symbol = "La", .name = "Lanthanum", .ano = 58 , .am =138.90 , .type = "Lanthanide", .block = 'f', .orbits = 6, .state = 's'},

    {.symbol = "Ce", .name = "Cerium", .ano = 59, .am =140.11 , .type = "Lanthanide", .block = 'f', .orbits = 6, .state = 's'},

    {.symbol = "Pr", .name = "Prasmodymium", .ano = 60, .am =140.90 , .type = "Lanthanide", .block = 'f', .orbits = 6, .state = 's'},

    {.symbol = "Nd", .name = "Neodymium", .ano = 61, .am = 144.24, .type = "Lanthanide", .block = 'f', .orbits = 6, .state = 's'},

    {.symbol = "Pm", .name = "Promethium", .ano = 62, .am =145 , .type = "Lanthanide", .block = 'f', .orbits = 6, .state = 's'},

    {.symbol = "Sm", .name = "Samarium", .ano = 63, .am =150.36 , .type = "Lanthanide", .block = 'f', .orbits = 6, .state = 's'},

    {.symbol = "Eu", .name = "Europium", .ano = 64, .am =151.96 , .type = "Lanthanide", .block = 'f', .orbits = 6, .state = 's'},

    {.symbol = "Gd", .name = "Gadolinium", .ano = 65, .am =157.25 , .type = "Lanthanide", .block = 'f', .orbits = 6, .state = 's'},

    {.symbol = "Tb", .name = "Terbium", .ano = 66, .am = 152.92, .type = "Lanthanide", .block = 'f', .orbits = 6, .state = 's'},

    {.symbol = "Dy", .name = "Dysprosium", .ano = 67, .am =162.50 , .type = "Lanthanide", .block = 'f', .orbits = 6, .state = 's'},

    {.symbol = "Ho", .name = "Holmium", .ano = 68, .am =164.93 , .type = "Lanthanide", .block = 'f', .orbits = 6, .state = 's'},

    {.symbol = "Er", .name = "Erbium", .ano = 69, .am = 167.25, .type = "Lanthanide", .block = 'f', .orbits = 6, .state = 's'},

    {.symbol = "Tm", .name = "Thulium", .ano = 70, .am = 168.93, .type = "Lanthanide", .block = 'f', .orbits = 6, .state = 's'},

    {.symbol = "Yb", .name = "Ytterbium", .ano =71, .am = 173.04, .type = "Lanthanide", .block = 'f', .orbits = 6, .state = 's'},

    {.symbol = "Lu", .name = "Lutetium", .ano = 72, .am =174.96 , .type = "Lanthanide", .block = 'f', .orbits = 6, .state = 's'},
```

```
    {.symbol = "Ta", .name = "Tantalum", .ano = 73, .am = 180.95, .type = "Transition Metal", .block = 'd', .orbits = 6, .state = 's'},

    {.symbol = "W", .name = "Tungesten", .ano = 74, .am = 183.84, .type = "Transition Metal", .block = 'd', .orbits = 6, .state = 's'},

    {.symbol = "Re", .name = "Rhenium", .ano = 75, .am = 186.21, .type = "Transition Metal", .block = 'd', .orbits = 6, .state = 's'},

    {.symbol = "Os", .name = "Osmium", .ano = 76, .am = 190.23, .type = "Transition Metal", .block = 'd', .orbits = 6, .state = 's'},

    {.symbol = "Ir", .name = "Iridium", .ano = 77, .am = 192.22, .type = "Transition Metal", .block = 'd', .orbits = 6, .state = 's'},

    {.symbol = "Pt", .name = "Platinum", .ano = 78, .am = 195.08, .type = "Transition Metal", .block = 'd', .orbits = 6, .state = 's'},

    {.symbol = "Au", .name = "Gold", .ano = 79, .am = 196.97, .type = "Transition Metal", .block = 'd', .orbits = 6, .state = 's'},

    {.symbol = "Hg", .name = "Mercury", .ano = 80, .am = 200.59, .type = "Post-Transition Metal", .block = 'd', .orbits = 6, .state = 'l'},

    {.symbol = "Tl", .name = "Thallium", .ano = 81, .am = 204.38, .type = "Post-Transition Metal", .block = 'p', .orbits = 6, .state = 's'},

    {.symbol = "Pb", .name = "Lead", .ano = 82, .am = 207.2, .type = "Post-Transition Metal", .block = 'p', .orbits = 6, .state = 's'},

    {.symbol = "Bi", .name = "Bismuth", .ano = 83, .am = 208.98, .type = "Post-Transition Metal", .block = 'p', .orbits = 6, .state = 's'},

    {.symbol = "Po", .name = "Polonium", .ano = 84, .am = 209, .type = "Post-Transition Metal", .block = 'p', .orbits = 6, .state = 's'},

    {.symbol = "At", .name = "Astatine", .ano = 85, .am = 210, .type = "Halogen", .block = 'p', .orbits = 6, .state = 's'},

    {.symbol = "Rn", .name = "Radon", .ano = 86, .am = 222, .type = "Noble Gas", .block = 'p', .orbits = 6, .state = 'g'},

    {.symbol = "Fr", .name = "Francium", .ano = 87, .am = 223, .type = "Alkali Metal", .block = 's', .orbits = 7, .state = 's'},

    {.symbol = "Ra", .name = "Radium", .ano = 88, .am = 226, .type = "Alkaline Earth Metal", .block = 's', .orbits = 7, .state = 's'},

    {.symbol = "Ac", .name = "Actinium", .ano = 89, .am = 227, .type = "Actinide", .block = 'd', .orbits = 7, .state = 's'},

    {.symbol = "Th", .name = "Thorium", .ano = 90, .am = 232.04, .type = "Actinide", .block = 'f', .orbits = 7, .state = 's'},
```

```
    {.symbol = "Pa", .name = "Protactinium", .ano = 91, .am = 231.04, .type = "Actinide",
.block = 'f', .orbits = 7, .state = 's'},

    {.symbol = "U", .name = "Uranium", .ano = 92, .am = 238.03, .type = "Actinide", .block =
'f', .orbits = 7, .state = 's'},

    {.symbol = "Np", .name = "Neptunium", .ano = 93, .am = 237, .type = "Actinide", .block =
'f', .orbits = 7, .state = 's'},

    {.symbol = "Pu", .name = "Plutonium", .ano = 94, .am = 244, .type = "Actinide", .block =
'f', .orbits = 7, .state = 's'},

    {.symbol = "Am", .name = "Americium", .ano = 95, .am = 243, .type = "Actinide", .block
= 'f', .orbits = 7, .state = 's'},

    {.symbol = "Cm", .name = "Curium", .ano = 96, .am = 247, .type = "Actinide", .block = 'f',
.orbits = 7, .state = 's'},

    {.symbol = "Bk", .name = "Berkelium", .ano = 97, .am = 247, .type = "Actinide", .block =
'f', .orbits = 7, .state = 's'},

    {.symbol = "Cf", .name = "Californium", .ano = 98, .am = 251, .type = "Actinide", .block
= 'f', .orbits = 7, .state = 's'},

    {.symbol = "Es", .name = "Einsteinium", .ano = 99, .am = 252, .type = "Actinide", .block =
'f', .orbits = 7, .state = 's'},

    {.symbol = "Fm", .name = "Fermium", .ano = 100, .am = 257, .type = "Actinide", .block =
'f', .orbits = 7, .state = 's'},

    {.symbol = "Md", .name = "Mendelevium", .ano = 101, .am = 258, .type = "Actinide",
.block = 'f', .orbits = 7, .state = 's'},

    {.symbol = "No", .name = "Nobelium", .ano = 102, .am = 259, .type = "Actinide", .block =
'f', .orbits = 7, .state = 's'},

    {.symbol = "Lr", .name = "Lawrencium", .ano = 103, .am = 266, .type = "Actinide", .block
= 'f', .orbits = 7, .state = 's'},

    {.symbol = "Rf", .name = "Rutherfordium", .ano = 104, .am = 267, .type = "Transition
Metal", .block = 'd', .orbits = 7, .state = 's'},

    {.symbol = "Db", .name = "Dubnium", .ano = 105, .am = 268, .type = "Transition Metal",
.block = 'd', .orbits = 7, .state = 's'},

    {.symbol = "Sg", .name = "Seaborgium", .ano = 106, .am = 269, .type = "Transition
Metal", .block = 'd', .orbits = 7, .state = 's'},

    {.symbol = "Bh", .name = "Bohrium", .ano = 107, .am = 270, .type = "Transition Metal",
.block = 'd', .orbits = 7, .state = 's'},

    {.symbol = "Hs", .name = "Hassium", .ano = 108, .am = 277, .type = "Transition Metal",
.block = 'd', .orbits = 7, .state = 's'},
```

```
    {.symbol = "Mt", .name = "Meithnerium", .ano = 109, .am = 278, .type = "Transition
Metal", .block = 'd', .orbits = 7, .state = 's'},

    {.symbol = "Ds", .name = "Darmstadtium", .ano = 110, .am = 281, .type = "Transition
Metal", .block = 'd', .orbits = 7, .state = 's'},

    {.symbol = "Rg", .name = "Roentgenium", .ano = 111, .am = 282, .type = "Transition
Metal", .block = 'd', .orbits = 7, .state = 's'},

    {.symbol = "Cn", .name = "Copernicium", .ano = 112, .am = 285, .type = "Transition
Metal", .block = 'd', .orbits = 7, .state = 'g'},

    {.symbol = "Nh", .name = "Nihonium", .ano = 113, .am = 286, .type = "Post-Transition
Metal", .block = 'p', .orbits = 7, .state = 's'},

    {.symbol = "Fl", .name = "Flerovium", .ano = 114, .am = 289, .type = "Post-Transition
Metal", .block = 'p', .orbits = 7, .state = 'g'},

    {.symbol = "Mc", .name = "Moscovium", .ano = 115, .am = 290, .type = "Post-Transition
Metal", .block = 'p', .orbits = 7, .state = 's'},

    {.symbol = "Lv", .name = "Livermorium", .ano = 116, .am = 293, .type = "Post-Transition
Metal", .block = 'p', .orbits = 7, .state = 's'},

    {.symbol = "Ts", .name = "Tennessine", .ano = 117, .am = 294, .type = "Halogen", .block
= 'p', .orbits = 7, .state = 's'},

    {.symbol = "Og", .name = "Oganesson", .ano = 118, .am = 294, .type = "Nobel Gas",
.block = 'p', .orbits = 7, .state = 's'}};

    //declaring all the variables

    char in[3];

    int d=1,i,j,k,y,exist,opt,f,n;

    float radius,velocity,force,density;

    struct ptable *p;

    //main menu

    while(d==1)

    {

    printf("\nChoose one of the following:\n1.Know about a particular element\n2.Alkali
Metals\n3.Alkaline Earth Metals\n4.Transition Metals\n5.Post-Transition
Metals\n6.Metalloids\n7.Non-Metals(except for halogens)\n8.Halogens\n9.Noble
Gases\n10.Lanthanides\n11.Actinindes\n12.Elements in s block\n13.Elements in p
block\n14.Elements in d block\n15.Elements in f block\n16.Elements that are solid at room
temperature\n17.Elements that are liquid at room temperature\n18.Elements that are gaseous
at room temperature\n");

    printf("\nEnter choice: ");
```

```
y=0;
scanf("%d",&y);
//about particular element
if(y==1)
{
while(y==1)
{
   printf("\nEnter name of element: ");
   scanf("%s",in);
   exist=0;
   //pointer initiation for particular element's structure
   for(i=0;i<=117;i++)
   {
      k=1;
      if(strlen(in)==strlen(elements[i].symbol))
      {
      for(j=0;j<strlen(in);j++)
      {
         if(tolower(in[j])!=tolower(elements[i].symbol[j]))
         {
            k=0;
            break;
         }
      }
      //assigning address to pointer for specific element
      if(k==1)
      {
         p=&elements[i];
         exist=1;
         break;
```

```
        }
        }
    }
    //functionalities
    if(exist==1)
    {
        printf("\nAbout the element:\nSymbol: %s\nName: %s\nAtomic number: %d\nAtomic Mass: %f\nType: %s\nBlock: %c\nNumber of orbits: %d\nState at room temperature: %c\n",p->symbol,p->name,p->ano,p->am,p->type,p->block,p->orbits,p->state);
        printf("\nChoose a property that you want to see (according to Bohr's formulas):\n1.Nuclei Radius\n2.Orbital Radius\n3.Electron Velocity\n4.Force needed to remove electron from atom\n5.Density of atom\n6.Density of nucleus\n");
    }
    else
        printf("This element does not exist.\n");
    opt=0;
    f=1;
    //properties
    while((f==1)&&(exist==1))
    {
        printf("\nenter choice: ");
        scanf("%d",&opt);
        //nucleus radius
        if(opt==1)
        {
            radius = (pow((p->am),0.33))*1.2;
            printf("Nuclei Radius of %s is: %fx10^-15 metres\n",p->name,radius);
        }
        //orbit radius
        else if(opt==2)
        {
            int n;
```

```
            printf("enter orbit number: ");
            scanf("%d",&n);
            if((0<n)&&(n<=p->orbits))
            {
               radius = 0.529*n/(p->ano)/(p->ano);
               printf("Orbital radius of orbit number %d of the element %s is: %f
Armstrong\n",n,p->name,radius);
            }
            else
            {
               printf("%s does not have this orbit in its ground state\n",p->name);
            }
         }
         //electron velocity
         else if(opt==3)
         {
            int n;
            printf("enter orbit number in which the electron is: ");
            scanf("%d",&n);
            if((0<n)&&(n<=p->orbits))
            {
               velocity = 2.18*pow(10,6)*(p->ano)/n;
               printf("Velocity of an electron in the orbit %d of the element %s is: %f
metres/second\n",n,p->name,velocity);
            }
            else
            {
               printf("%s does not have this orbit in its ground state\n",p->name);
            }
         }
         //electrostatic force
```

```
else if(opt==4)
{
   printf("enter orbit number in which the electron is: ");
   scanf("%d",&n);
   if((0<n)&&(n<=p->orbits))
   {
      force = 9*1.6*1.6*pow(p->ano,3)/pow(0.529*n,2);
      printf("Force needed to remove a electron in the orbit %d from an atom of the element %s is: %fx10^-9 Newton\n",n,p->name,force);
   }
   else
   {
      printf("%s does not have this orbit in its ground state\n",p->name);
   }
}
//atomic density
else if(opt==5)
{
   density = (p->am)*1.66/(4*3.14*pow((0.529*(p->orbits)/(p->ano)/(p->ano)),3)/3);
   printf("Density of %s is: %fx10^3 kilogram/cubic metre\n",p->name,density);
}
//nucleus density
else if(opt==6)
{
   density = 1.66/(4*3.14*pow(1.2,3)/3);
   printf("Density of the nucleus is: %fx10^18 kilogram/cubic metre\nNOTE: This value is same for all the atoms.\n",density);
}
//invalid choice
else
{
```

```
            printf("That choice is not available.");
        }
        //to see other properties
        printf("\nDo you want to explore the properties more? Please enter 1(yes) or 0(no): ");
        scanf("%d",&f);
      }
      printf("We hope the element was interesting enough for you !!\n");
      //to repeat for another element
      printf("\nDo you want to know about some other element? Please enter 1(yes) or 0(no):
");
      scanf("%d",&y);
   }
   printf("If you want to know any more about elements please GOOGLE..\n");
   }
   //alkali metals
   else if(y==2)
   {
      printf("Alkali Metals:\n");
      char a[14]="Alkali Metal";
      for(i=0;i<118;i++)
      {
         k=1;
         for(j=0;j<13;j++)
         {
            if(a[j]!=elements[i].type[j])
            {
               k=0;
               break;
            }
         }
         if(k==1)
```

```
        {
            printf("%s\n",elements[i].name);
        }
    }
}
//alkaline earth metals
else if(y==3)
{
    printf("Alkaline Earth Metals:\n");
    char a[22]="Alkaline Earth Metal";
    for(i=0;i<118;i++)
    {
        k=1;
        for(j=0;j<21;j++)
        {
            if(a[j]!=elements[i].type[j])
            {
                k=0;
                break;
            }
        }
        if(k==1)
        {
            printf("%s\n",elements[i].name);
        }
    }
}
//transition metals
else if(y==4)
{
```

```
    printf("Transition Metals:\n");
    char a[17]="Transition Metal";
    for(i=0;i<118;i++)
    {
       k=1;
       for(j=0;j<15;j++)
       {
          if(a[j]!=elements[i].type[j])
          {
             k=0;
             break;
          }
       }
       if(k==1)
       {
          printf("%s\n",elements[i].name);
       }
    }
 }
 //post-transition metals
 else if(y==5)
 {
    printf("Post-Transition Metals:\n");
    char a[22]="Post-Transition Metal";
    for(i=0;i<118;i++)
    {
       k=1;
       for(j=0;j<21;j++)
       {
          if(a[j]!=elements[i].type[j])
```

```
            {
                k=0;
                break;
            }
        }
        if(k==1)
        {
            printf("%s\n",elements[i].name);
        }
    }
}
//metalloids
else if(y==6)
{
    printf("Mettalloids:\n");
    char a[10]="Metalloid";
    for(i=0;i<118;i++)
    {
        k=1;
        for(j=0;j<9;j++)
        {
            if(a[j]!=elements[i].type[j])
            {
                k=0;
                break;
            }
        }
        if(k==1)
        {
            printf("%s\n",elements[i].name);
```

```
        }
    }
}
//non-metals
else if(y==7)
{
    printf("Non-Metals (except halogens):\n");
    char a[10]="Non-Metal";
    for(i=0;i<118;i++)
    {
        k=1;
        for(j=0;j<9;j++)
        {
            if(a[j]!=elements[i].type[j])
            {
                k=0;
                break;
            }
        }
        if(k==1)
        {
            printf("%s\n",elements[i].name);
        }
    }
}
//halogens
else if(y==8)
{
    printf("Halogens:\n");
    char a[8]="Halogen";
```

```
    for(i=0;i<118;i++)
    {
        k=1;
        for(j=0;j<7;j++)
        {
            if(a[j]!=elements[i].type[j])
            {
                k=0;
                break;
            }
        }
        if(k==1)
        {
            printf("%s\n",elements[i].name);
        }
    }
}
//noble gases
else if(y==9)
{
    printf("Noble Gases:\n");
    char a[10]="Noble Gas";
    for(i=0;i<118;i++)
    {
        k=1;
        for(j=0;j<9;j++)
        {
            if(a[j]!=elements[i].type[j])
            {
                k=0;
```

```
                break;
            }
        }
        if(k==1)
        {
            printf("%s\n",elements[i].name);
        }
    }
}
//lanthanides
else if(y==10)
{
    printf("Lanthanides:\n");
    char a[11]="Lanthanide";
    for(i=0;i<118;i++)
    {
        k=1;
        for(j=0;j<10;j++)
        {
            if(a[j]!=elements[i].type[j])
            {
                k=0;
                break;
            }
        }
        if(k==1)
        {
            printf("%s\n",elements[i].name);
        }
    }
```

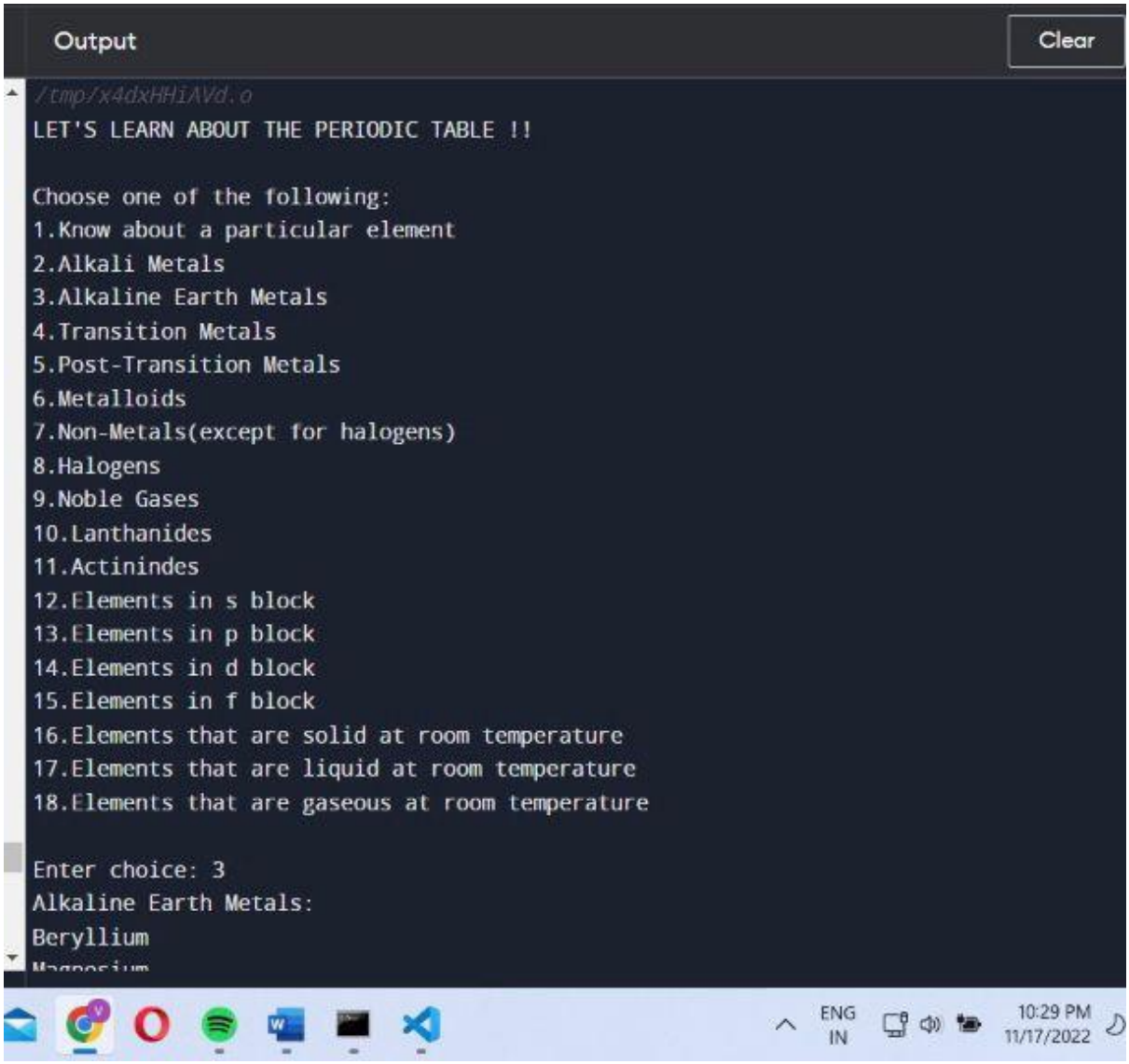
```
    }
    //actinides
    else if(y==11)
    {
        printf("Actinides:\n");
        char a[9]="Actinide";
        for(i=0;i<118;i++)
        {
            k=1;
            for(j=0;j<8;j++)
            {
                if(a[j]!=elements[i].type[j])
                {
                    k=0;
                    break;
                }
            }
            if(k==1)
            {
                printf("%s\n",elements[i].name);
            }
        }
    }
    //s block
    else if(y==12)
    {
        printf("S Block Elements:\n");
        for(i=0;i<118;i++)
        {
            if(elements[i].block=='s')
```

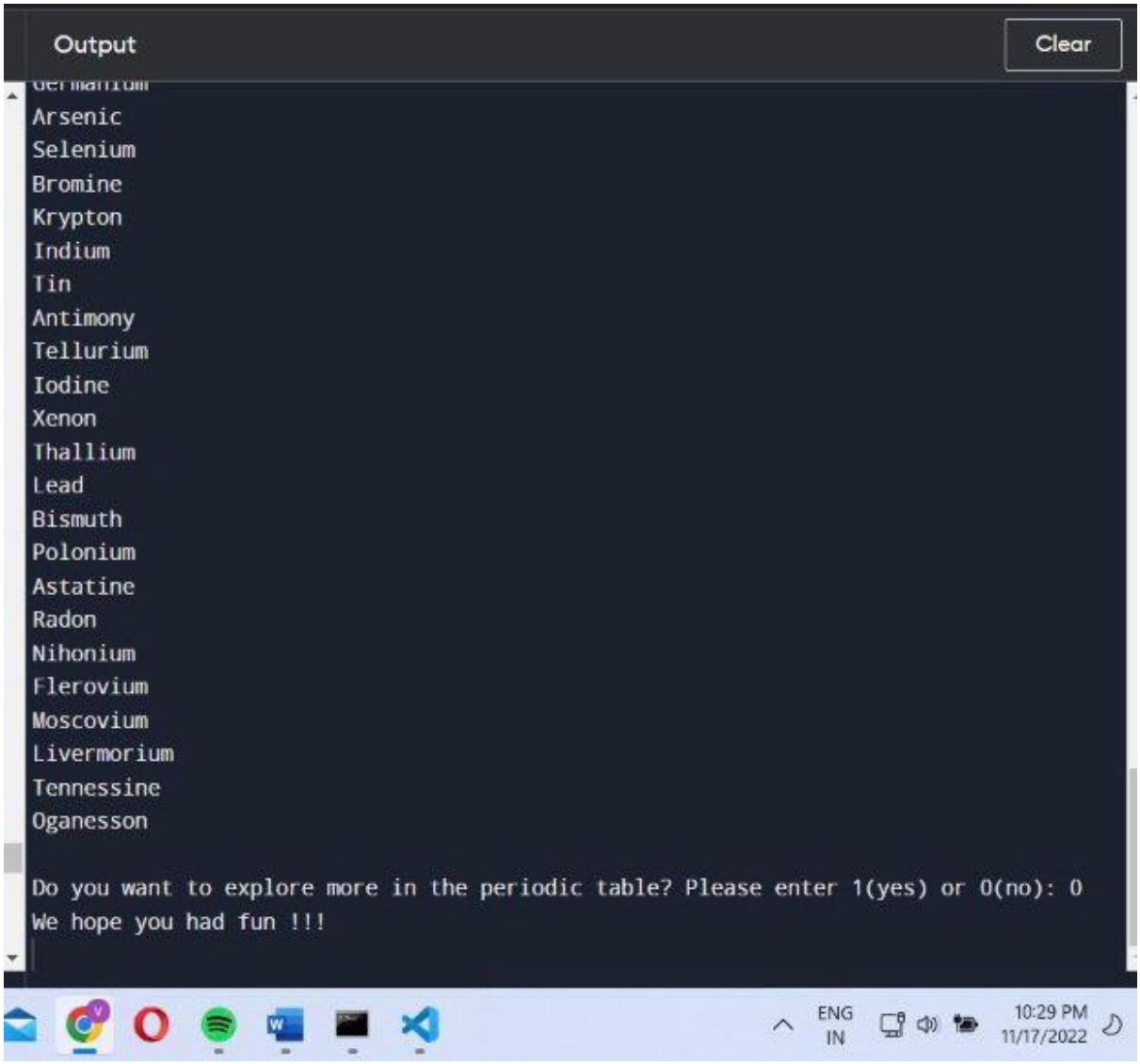
```
        {
            printf("%s\n",elements[i].name);
        }
    }
}
//p block
else if(y==13)
{
    printf("P Block Elements:\n");
    for(i=0;i<118;i++)
    {
        if(elements[i].block=='p')
        {
            printf("%s\n",elements[i].name);
        }
    }
}
//d block
else if(y==14)
{
    printf("D Block Elements:\n");
    for(i=0;i<118;i++)
    {
        if(elements[i].block=='d')
        {
            printf("%s\n",elements[i].name);
        }
    }
}
//f block
```

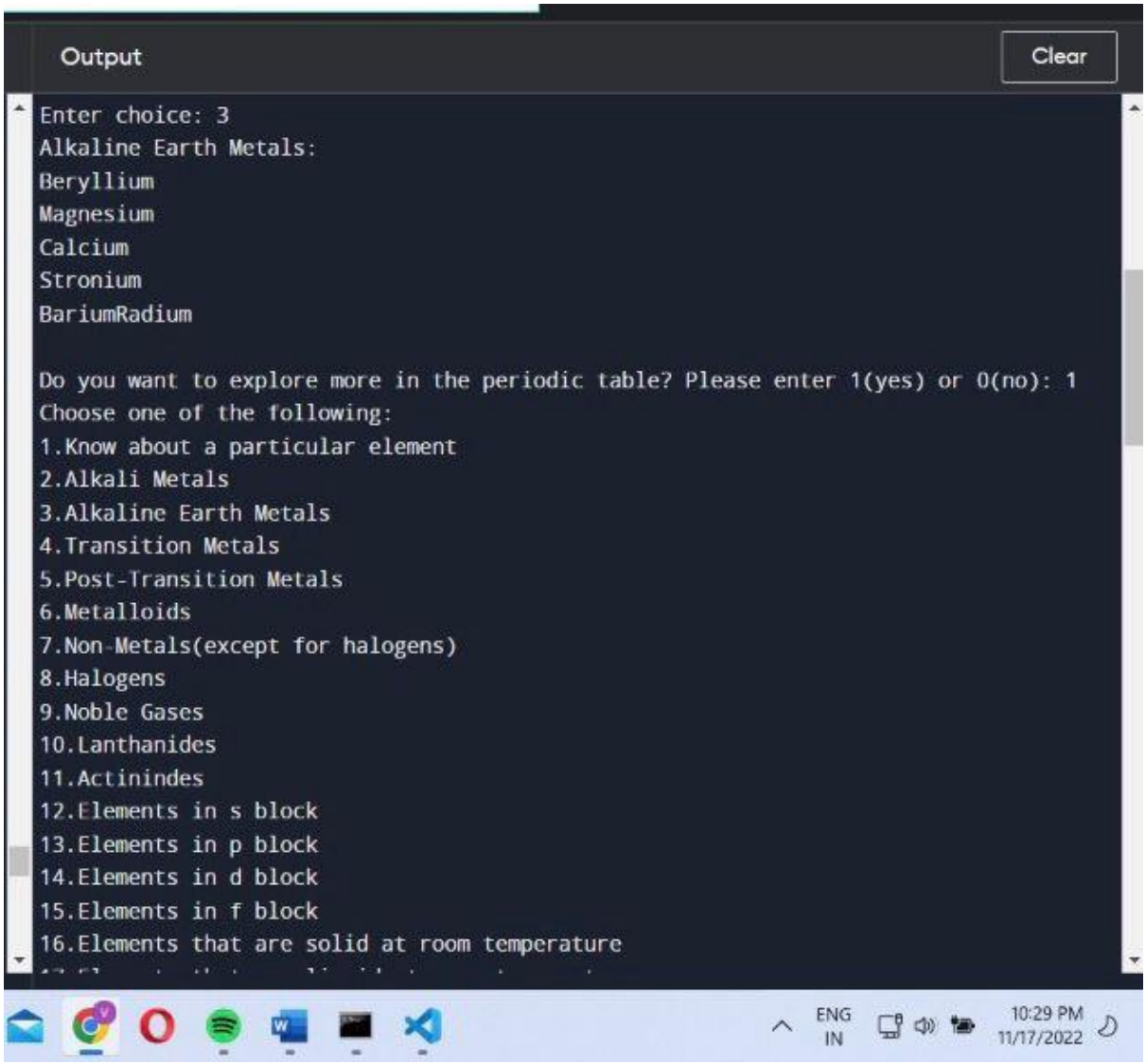
```
else if(y==15)
{
   printf("F Block Elements:\n");
   for(i=0;i<118;i++)
   {
      if(elements[i].block=='f')
      {
         printf("%s\n",elements[i].name);
      }
   }
}
//solid
else if(y==16)
{
   printf("Elements that are solid at room temperature:\n");
   for(i=0;i<118;i++)
   {
      if(elements[i].state=='s')
      {
         printf("%s\n",elements[i].name);
      }
   }
}
//liquid
else if(y==17)
{
   printf("Elements that are liquid at room temperature:\n");
   for(i=0;i<118;i++)
   {
      if(elements[i].state=='l')
```

```
            {
                printf("%s\n",elements[i].name);
            }
        }
    }
    //gaseous
    else if(y==18)
    {
        printf("Elements that are gaseous at room temperature:\n");
        for(i=0;i<118;i++)
        {
            if(elements[i].state=='g')
            {
                printf("%s\n",elements[i].name);
            }
        }
    }
    //invalid choice
    else
    {
        printf("This choice is not available.\n");
    }
    //to go back to the start menu if the user want to try more options
    printf("\nDo you want to explore more in the periodic table? Please enter 1(yes) or 0(no):
");
    scanf("%d",&d);
    }
    printf("We hope you had fun !!!\n");
}
```

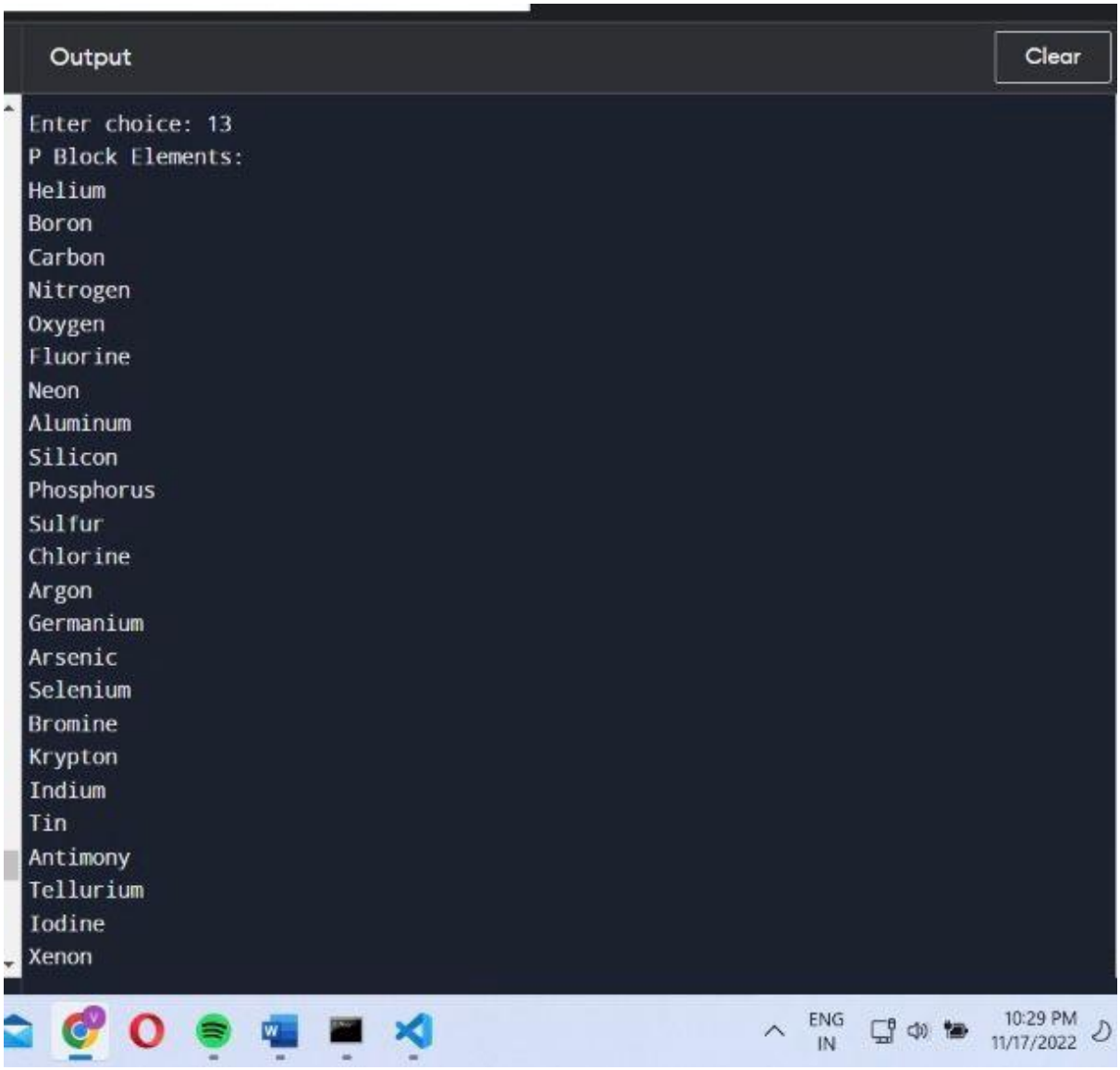
Output:

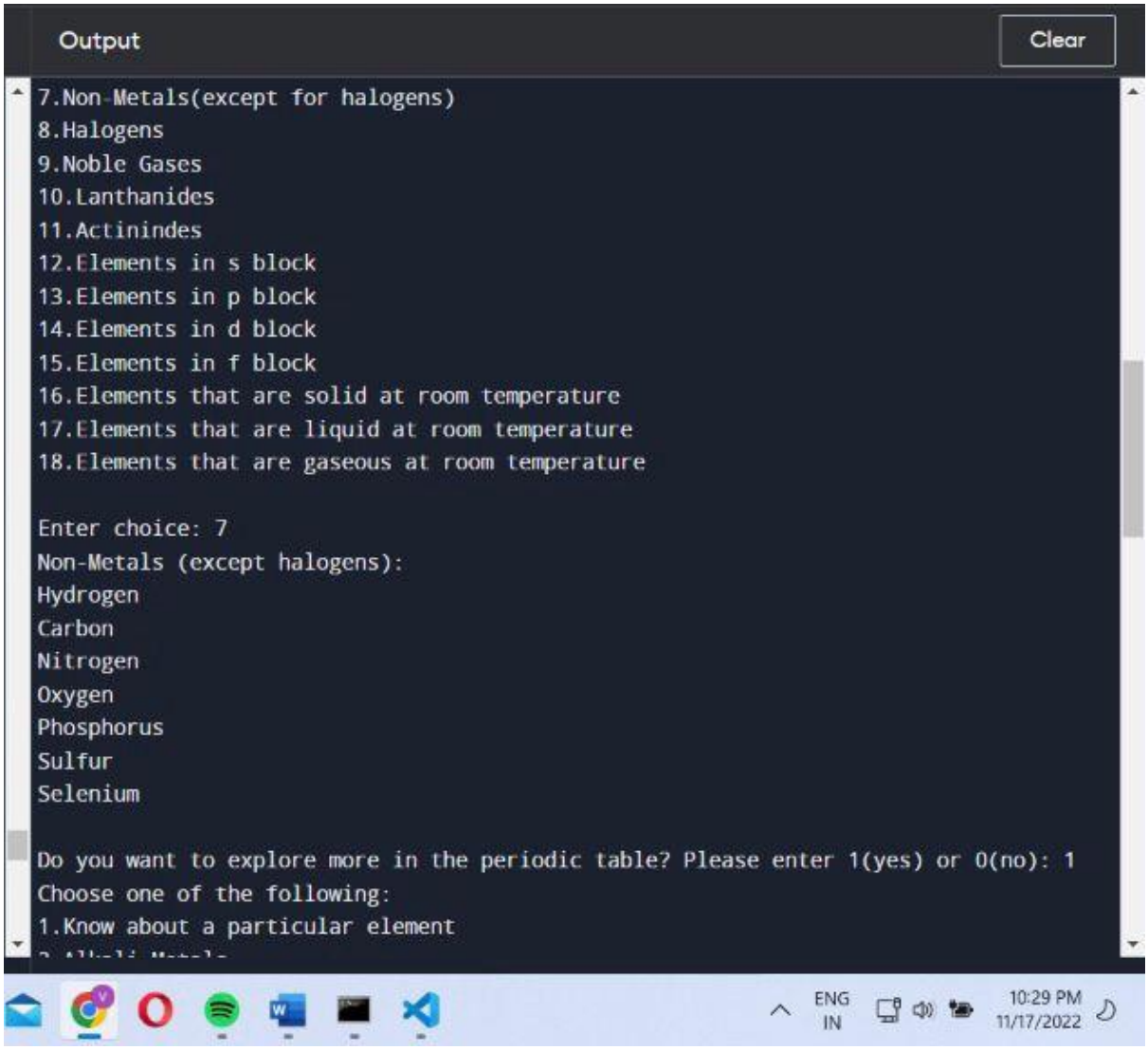

Output
Clear
/tmp/x4dxHHiAVd.o
LET'S LEARN ABOUT THE PERIODIC TABLE !!
Choose one of the following:
1.Know about a particular element
2.Alkali Metals
3.Alkaline Earth Metals
4.Transition Metals
5.Post-Transition Metals
6.Metalloids
7.Non-Metals(except for halogens)
8.Halogens
9.Noble Gases
10.Lanthanides
11.Actinindes
12.Elements in s block
13.Elements in p block
14.Elements in d block
15.Elements in f block
16.Elements that are solid at room temperature
17.Elements that are liquid at room temperature
18.Elements that are gaseous at room temperature
Enter choice: 3
Alkaline Earth Metals:
Beryllium
ENG
IN
10:29 PM
11/17/2022

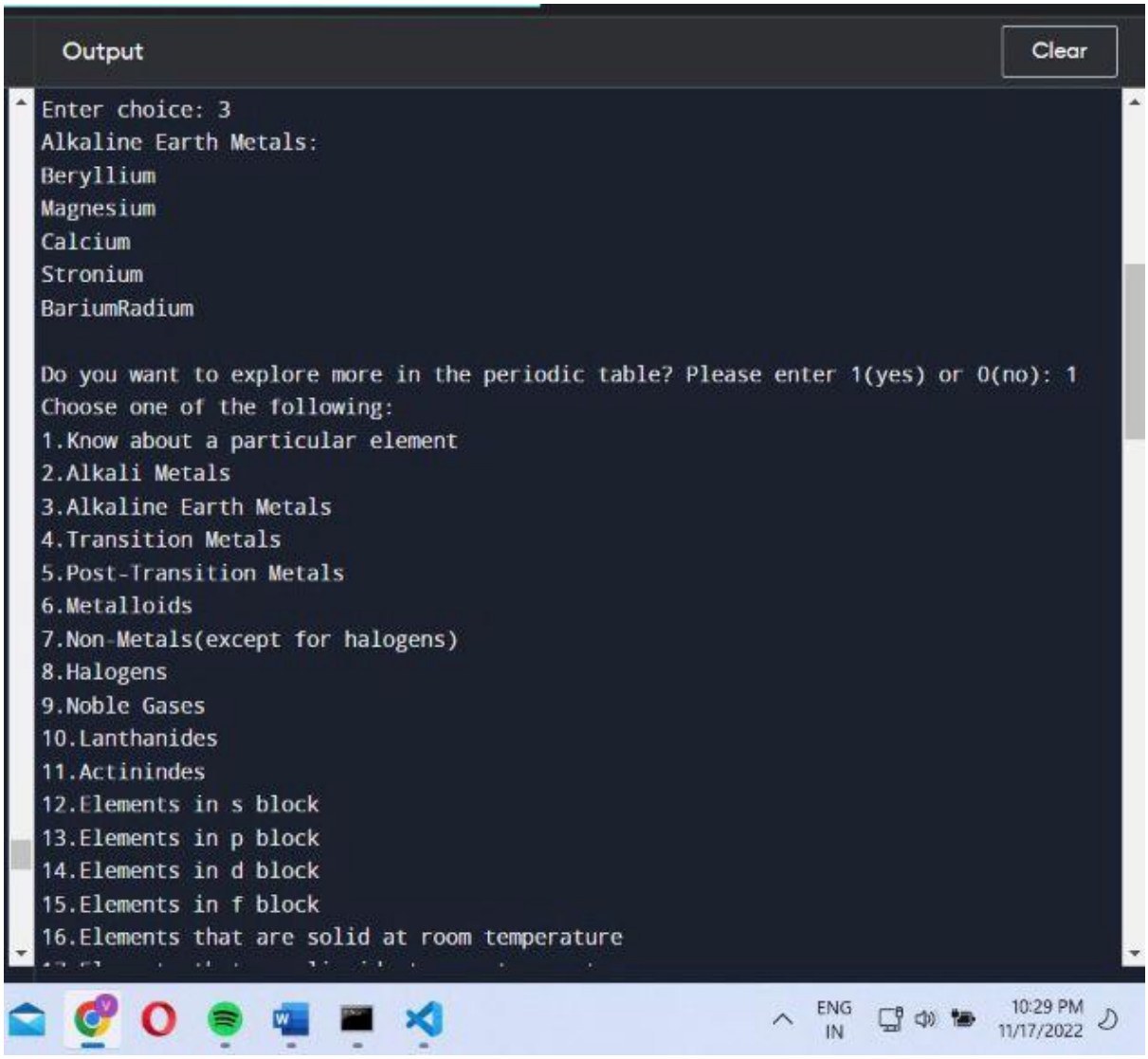

Output
Clear
Arsenic
Selenium
Bromine
Krypton
Indium
Tin
Antimony
Tellurium
Iodine
Xenon
Thallium
Lead
Bismuth
Polonium
Astatine
Radon
Nihonium
Flerovium
Moscovium
Livermorium
Tennessine
Oganesson
Do you want to explore more in the periodic table? Please enter 1(yes) or 0(no): 0
We hope you had fun !!!
ENG
IN
10:29 PM
11/17/2022


Output
Clear
Enter choice: 3
Alkaline Earth Metals:
Beryllium
Magnesium
Calcium
Stronium
BariumRadium
Do you want to explore more in the periodic table? Please enter 1(yes) or 0(no): 1
Choose one of the following:
1.Know about a particular element
2.Alkali Metals
3.Alkaline Earth Metals
4.Transition Metals
5.Post-Transition Metals
6.Metalloids
7.Non-Metals(except for halogens)
8.Halogens
9.Noble Gases
10.Lanthanides
11.Actinindes
12.Elements in s block
13.Elements in p block
14.Elements in d block
15.Elements in f block
16.Elements that are solid at room temperature
ENG
IN
10:29 PM
11/17/2022


Output
Clear
Do you want to explore more in the periodic table? Please enter 1(yes) or 0(no): 1
Choose one of the following:
1.Know about a particular element
2.Alkali Metals
3.Alkaline Earth Metals
4.Transition Metals
5.Post-Transition Metals
6.Metalloids
7.Non-Metals(except for halogens)
8.Halogens
9.Noble Gases
10.Lanthanides
11.Actinindes
12.Elements in s block
13.Elements in p block
14.Elements in d block
15.Elements in f block
16.Elements that are solid at room temperature
17.Elements that are liquid at room temperature
18.Elements that are gaseous at room temperature
Enter choice: 13
P Block Elements:
Helium
Boron
Carbon
ENG
IN
10:29 PM
11/17/2022


Output
Clear
Enter choice: 13
P Block Elements:
Helium
Boron
Carbon
Nitrogen
Oxygen
Fluorine
Neon
Aluminum
Silicon
Phosphorus
Sulfur
Chlorine
Argon
Germanium
Arsenic
Selenium
Bromine
Krypton
Indium
Tin
Antimony
Tellurium
Iodine
Xenon
ENG
IN
10:29 PM
11/17/2022


Output
Clear
7.Non-Metals(except for halogens)
8.Halogens
9.Noble Gases
10.Lanthanides
11.Actinindes
12.Elements in s block
13.Elements in p block
14.Elements in d block
15.Elements in f block
16.Elements that are solid at room temperature
17.Elements that are liquid at room temperature
18.Elements that are gaseous at room temperature
Enter choice: 7
Non-Metals (except halogens):
Hydrogen
Carbon
Nitrogen
Oxygen
Phosphorus
Sulfur
Selenium
Do you want to explore more in the periodic table? Please enter 1(yes) or 0(no): 1
Choose one of the following:
1.Know about a particular element
ENG
IN
10:29 PM
11/17/2022


Output
Clear
Enter choice: 3
Alkaline Earth Metals:
Beryllium
Magnesium
Calcium
Stronium
BariumRadium
Do you want to explore more in the periodic table? Please enter 1(yes) or 0(no): 1
Choose one of the following:
1.Know about a particular element
2.Alkali Metals
3.Alkaline Earth Metals
4.Transition Metals
5.Post-Transition Metals
6.Metalloids
7.Non-Metals(except for halogens)
8.Halogens
9.Noble Gases
10.Lanthanides
11.Actinindes
12.Elements in s block
13.Elements in p block
14.Elements in d block
15.Elements in f block
16.Elements that are solid at room temperature
ENG
IN
10:29 PM
11/17/2022

Project 2: Stack

SPECIAL VENDING MACHINE

Code:

```
//task - create a program that uses stack with real-life applications
//SPECIAL VENDING MACHINE

#include<stdio.h>
#include<string.h>

//structure for info on the food options
struct vm //vending machine - vm
{
   char stack[10][10];  //stack of the packets
   char name[60];
   int price;
   char taste[25];
   char category[25];
   int tos; //top of stack - initially 4 i.e. 5 packets
};

//push function - KRUPA
int push(char x[],char array[10][10],int Top)
{
        if(Top==10-1)
        {
                printf("\nOverflow!!");
                return 9;
        }
        Top+=1;
```

```
        int i;
        for(i=0;i<strlen(x);i++)
      array[Top][i]=x[i];
   return Top;
}

//pop function - KRUPA
int pop(char array[10][10],int Top)
{
   if(Top==-1)
        {
              printf("\nUnderflow!!");
              return -1;
        }
   return --Top;
}

//function to display according to categories - VRUNDA
void catgy(struct vm food[])
{
   int op = 1, i, k, j, choice;
   char in_cat[50] = "";
   printf("\n1-Lays\n2-Kurkure\n3-Chocolate\n4-Drinks\n5-Namkeen\n6-Khakra\n7-
Nachos\n8-Biscuits");
   while (op != 0)
   {
      printf("\n\nIf you want to know about the variety available then press 1(YES) else press
0(NO): ");
      scanf("%d",&op);
      if (op == 1)
      {
```

```
printf("\nPlease choose a number to know about a particular variety:");
scanf("%d",&choice);
switch(choice)
{
   case  1:
      strcpy(in_cat, "Lays");
      break;
   case  2:
      strcpy(in_cat, "Kurkure");
      break;
   case  3:
      strcpy(in_cat, "Chocolate");
      break;
   case  4:
      strcpy(in_cat, "Drinks");
      break;
   case  5:
      strcpy(in_cat, "Namkeen");
      break;
   case 6:
      strcpy(in_cat, "Khakra");
      break;
   case  7:
      strcpy(in_cat, "Nachos");
      break;
   case  8:
      strcpy(in_cat, "Biscuits");
      break;
   default:
      break;
```

```
        }
        for(i=0; i < 42; i++)
        {
            if (strcmp(food[i].category, in_cat) == 0)
                printf("\n%d. %s - price: %d, number of available packets:
%d",i+1,food[i].name,food[i].price,food[i].tos+1);
        }
        if((choice>8)||(choice<1))
            printf("invalid choice");
    }
    else if(op==0)
        printf("\nWe hope that was informative!\n");
    else
        printf("\nERROR: please enter 1 or 0!!\n");
  }
}

//function to display according to taste - MISARI
void taste(struct vm food[])
{
  int op = 1, i, k, j, choice;
  char in_taste[50] = "";
  printf("\n1-Spicy\n2-Sweet\n3-Chatpata\n4-Mint\n5-Salty\n6-Bitter\n7-Tangy");
  while (op != 0)
    {
    printf("\n\nIf you want to know about the variety available in a particular taste then
press 1(YES) else press 0(NO): ");
    scanf("%d",&op);
    if (op == 1)
    {
        printf("\nPlease enter choice for taste category:");
```

```
scanf("%d",&choice);
switch(choice)
{
   case  1:
      strcpy(in_taste, "spicy");
      break;
   case  2:
      strcpy(in_taste, "sweet");
      break;
   case  3:
      strcpy(in_taste, "chatpata");
      break;
   case  4:
      strcpy(in_taste, "mint");
      break;
   case  5:
      strcpy(in_taste, "salty");
      break;
   case 6:
      strcpy(in_taste, "bitter");
      break;
   case  7:
      strcpy(in_taste, "tangy");
      break;
   default:
      break;
}
for(i=0; i < 42; i++)
{
   if (strcmp(food[i].taste, in_taste) == 0)
```

```
                printf("\n%d. %s - price: %d, number of available packets:
%d",i+1,food[i].name,food[i].price,food[i].tos+1);
        }
        if((choice>7)||(choice<1))
            printf("invalid choice");


    }
    else if(op==0)
        printf("\nWe hope that was informative!\n");
    else
        printf("\nERROR: please enter 1 or 0!!\n");
  }
}

//function to display according to price - TANYA
void price(struct vm t[])
{
        int d=1,p,i;
        while(d==1)
        {
                printf("\nAvailable price ranges:\n");
                printf("1.1-10\n2.11-20\n3.21-30\n4.31-40\n5.41-50\n6.51-60\n");
                printf("\nEnter Choice: ");
                scanf("%d",&p);
                if (p==1)
           {
                        for(i=0;i<42;i++)
                        {
                                if((t[i].price>=1)&&(t[i].price<=10))
                                        printf("\n%d. %s - price: %d, number of available
packets: %d",i+1,t[i].name,t[i].price,t[i].tos+1);
```

```
            }
        }
        else if(p==2)
        {
                for(i=0;i<42;i++)
                {
                        if((t[i].price>=11)&&(t[i].price<=20))
            printf("\n%d. %s - price: %d, number of available packets:
%d",i+1,t[i].name,t[i].price,t[i].tos+1);
                }
        }
        else if (p==3)
        {
                for(i=0;i<42;i++)
                {
                        if((t[i].price>=21)&&(t[i].price<=30))
            printf("\n%d. %s - price: %d, number of available packets:
%d",i+1,t[i].name,t[i].price,t[i].tos+1);
                }

        }
        else if (p==4)
        {
                for(i=0;i<42;i++)
                {
                        if((t[i].price>=31)&&(t[i].price<=40))
            printf("\n%d. %s - price: %d, number of available packets:
%d",i+1,t[i].name,t[i].price,t[i].tos+1);
                }
        }
        else if (p==5)
```

```
                {
                        for(i=0;i<42;i++)
                        {
                                if((t[i].price>=41)&&(t[i].price<=50))
                printf("\n%d. %s - price: %d, number of available packets:
%d",i+1,t[i].name,t[i].price,t[i].tos+1);
                        }
                }
                else if (p==6)
                {
                        for(i=0;i<42;i++)
                        {
                                if((t[i].price>=51)&&(t[i].price<=60))
                printf("\n%d. %s - price: %d, number of available packets:
%d",i+1,t[i].name,t[i].price,t[i].tos+1);
                        }
                }
                else
        printf("that choice is not available!!");
                printf("\n\nDo you want to see more price ranges? If yes enter 1 else enter any
integer: ");
                scanf("%d",&d);
        }
}


//main code -REHA
void main()
{
   printf("SPECIAL VENDING MACHINE\n\n");
   printf("Copyright ABCD 2022 and 1234567890\n\n");
   printf("-----------------------------------------------------------------\n");
```

```
//structure array for vm

struct vm food[42]={{.stack={"packet","packet","packet","packet","packet"," "," "," "," "
"},.name="Lays - Indian Magic
Masala",.price=30,.taste="chatpata",.category="Lays",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Lays -
Hot n Sweet Chilli",.price=20,.taste="spicy",.category="Lays",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Lays -
American Style Cream And Onion",.price=25,.taste="chatpata",.category="Lays",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Lays -
Spanish Tomato Tango",.price=40,.taste="spicy",.category="Lays",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Lays -
Chile Limon",.price=20,.taste="tangy",.category="Lays",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Lays -
Classic Salted",.price=40,.taste="salty",.category="Lays",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Kurkure -
Masala Munch",.price=20,.taste="chatpata",.category="Kurkure",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Kurkure -
Puff Corn",.price=20,.taste="chatpata",.category="Kurkure",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Kurkure -
Triangles",.price=20,.taste="salty",.category="Kurkure",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Kurkure -
Solid Masti",.price=20,.taste="chatpata",.category="Kurkure",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Kurkure -
Chataka Pataka",.price=20,.taste="chatpata",.category="Kurkure",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Kurkure -
Chili Chataka",.price=20,.taste="spicy",.category="Kurkure",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Nachos -
Peri Peri",.price=30,.taste="spicy",.category="Nachos",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Nachos -
Barbeque",.price=30,.taste="sweet",.category="Nachos",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Nachos -
Sizzlin Jalapeno",.price=30,.taste="spicy",.category="Nachos",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Nachos -
Tomato Mexican",.price=30,.taste="sweet-sour",.category="Nachos",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Nachos -
Tikka Masala",.price=30,.taste="spicy",.category="Nachos",.tos=4},
```

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Biscuits - Parle-G",.price=20,.taste="sweet",.category="Biscuits",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Biscuits - Bourbon",.price=30,.taste="sweet",.category="Biscuits",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Biscuits - Monaco",.price=10,.taste="salty",.category="Biscuits",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Biscuits - Jim Jam",.price=40,.taste="sweet",.category="Biscuits",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Biscuits - Good Day",.price=30,.taste="sweet",.category="Biscuits",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Khakra - Methi",.price=40,.taste="bitter",.category="Khakra",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Khakra - Masala",.price=50,.taste="spicy",.category="Khakra",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Khakra - Plain",.price=60,.taste="salty",.category="Khakra",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Khakra - Jeera",.price=30,.taste="salty",.category="Khakra",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Khakra - Chili",.price=55,.taste="spicy",.category="Khakra",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Chocolate - Dairy Milk",.price=10,.taste="sweet",.category="Chocolate",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Chocolate - KitKat",.price=25,.taste="sweet",.category="Chocolate",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Chocolate - Munch",.price=10,.taste="sweet",.category="Chocolate",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Chocolate - 5Star",.price=20,.taste="sweet",.category="Chocolate",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Chocolate - Amul Dark Chocolate",.price=80,.taste="bitter",.category="Chocolate",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Drinks - Thumbs-up",.price=40,.taste="tangy",.category="Drinks",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Drinks - Fanta",.price=40,.taste="sweet",.category="Drinks",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Drinks - Sprite",.price=40,.taste="tangy",.category="Drinks",.tos=4},

```
{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Drinks - Coca-cola",.price=40,.taste="tangy",.category="Drinks",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Drinks - Limca",.price=40,.taste="mint",.category="Drinks",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Namkeen - Bhakharvadi",.price=65,.taste="spicy",.category="Namkeen",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Namkeen - Aloo bhujiya",.price=35,.taste="chatpata",.category="Namkeen",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Namkeen - Lite mixture",.price=35,.taste="spicy",.category="Namkeen",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Namkeen - Salted peanuts",.price=50,.taste="chatpata",.category="Namkeen",.tos=4},

{.stack={"packet","packet","packet","packet","packet"," "," "," "," "," "},.name="Namkeen - Kachori",.price=60,.taste="spicy",.category="Namkeen",.tos=4},

};

//declaring all the variables

int f=1,p,opt,key,n,q,d,g,i,z;

//customer or manager

while(f==1)

{

   printf("\nAre you a:\n1.Customer\n2.Manager\n3.Or do you want to Leave application\n");

   printf("\nenter choice: ");

   scanf("%d",&p);

   //customer

   if(p==1)

   {

      z=1;

      g=1;

      while(g==1)

      {

         printf("\nchoose one of the following options to see available products:\n1.whole menu\n2.categorically filter\n3.filter on the basis of taste\n4.filter on the basis of the price range\n5.directly order\n6.don't want to order\n");
```

```
        printf("\nenter choice: ");
        scanf("%d",&opt);
        //whole menu
        if(opt==1)
        {
           printf("WHOLE MENU:\n)");
           for(i=0;i<42;i++)
              printf("\n%d. %s - price: %d, number of available packets:
%d",i+1,food[i].name,food[i].price,food[i].tos+1);
        }
        //categories
        else if(opt==2)
           catgy(food);
        //tastes
        else if(opt==3)
           taste(food);
        //price range
        else if(opt==4)
           price(food);
        //directly order
        else if(opt==5)
           break;
        //leave customer page
        else if(opt==6)
        {
           z=0;
           break;
        }
        //invalid
        else
           printf("choice is invalid.\n\n");
```

```
        //want to continue loop?
        printf("\nAre you ready to order - if not - enter 1 else enter any integer: ");
        scanf("%d",&g);
    }
    g=1;
    while((g==1)&&(z==1))
    {
        //item input
        d=1;
        while(d==1)
        {
            printf("\nEnter item number: ");
            scanf("%d",&n);
            if((n>0)&&(n<43))
                break;
            printf("invalid number, do you want to try again? press 1 if yes else enter any
integer: ");
            scanf("%d",&d);
        }
        //input amount of product
        while(d==1)
        {
            printf("enter quantity of packets: ");
            scanf("%d",&q);
            if(q<=(food[n-1].tos+1))
                break;
            printf("we don't have that much quantity, do you want to enter a smaller value?
(press 1 if yes, else press any integer): ");
            scanf("%d",&d);
        }
        //giving the products
```

```
            if(d==1)
            {
               for(i=0;i<q;i++)
                  food[n-1].tos=pop(food[n-1].stack,food[n-1].tos);
               printf("please pay %d rupees and collect your purchase(s): %d %s.\n",(food[n-1].price)*q,q,food[n-1].name);
            }
            //want to continue loop?
            printf("\nDo you want to purchase more? if yes, enter 1 else enter any integer: ");
            scanf("%d",&g);
         }
         printf("\nThank You for shopping with us!\nGood food brings good health :)\n");
      }
      //manager
      else if(p==2)
      {
         //admin key input
         g=1;
         while(g==1)
         {
            printf("\nenter admin pin: ");
            scanf("%d",&key);
            if(key==12624)
               break;
            printf("invalid key!!\nDo you want to enter again? press 1 if yes else any other integer: ");
            scanf("%d",&g);
         }
         //correct key
         while(g==1)
         {
```

```
            //item input
            d=1;
            while(d==1)
            {
               printf("\nEnter item number: ");
               scanf("%d",&n);
               if((n>0)&&(n<43))
                  break;
               printf("invalid number, do you want to try again? press 1 if yes else enter any
integer: ");
               scanf("%d",&d);
            }
            //input amount of product
            while(d==1)
            {
               printf("enter quantity of packets you are adding: ");
               scanf("%d",&q);
               if(q<=(9-food[n-1].tos))
                 break;
               printf("we don't have that much capacity, do you want to enter a smaller value?
(press 1 if yes, else press any integer): ");
               scanf("%d",&d);
            }
            //adding the packets
            if(d==1)
            {
               for(i=0;i<q;i++)
               {
                  food[n-1].tos=push("packet",food[n-1].stack,food[n-1].tos);
               }
```

```
                printf("\nSuccessfully added the packets!!\nUpdated number of packets of %s:
%d\n",food[n-1].name,food[n-1].tos+1);
            }
            //want to continue loop?
            printf("\nDo you want to add more packets? if yes, enter 1 else enter any integer:
");
            scanf("%d",&g);
        }
    }
    //leave
    else if(p==3)
    {
        printf("\nVENDING MACHINE SHUTDOWN.\n");
        f=0;
    }
    //invalid
    else
        printf("invalid choice\n");
  }
}
```

Output:

```
---------------------------------------------------------------------

Are you a:
1.Customer
2.Manager
3.Or do you want to Leave application

enter choice: 1

choose one of the following options to see available products:
1.whole menu
2.categorically filter
3.filter on the basis of taste
4.filter on the basis of the price range
5.directly order
6.don't want to order

enter choice: 1
WHOLE MENU:
)
1. Lays - Indian Magic Masala - price: 30, number of available packets: 5
2. Lays - Hot n Sweet Chilli - price: 20, number of available packets: 5
3. Lays - American Style Cream And Onion - price: 25, number of available packets: 5
4. Lays - Spanish Tomato Tango - price: 40, number of available packets: 5
5. Lays - Chile Limon - price: 20, number of available packets: 5
6. Lays - Classic Salted - price: 40, number of available packets: 5
7. Kurkure - Masala Munch - price: 20, number of available packets: 5
8. Kurkure - Puff Corn - price: 20, number of available packets: 5
9. Kurkure - Triangles - price: 20, number of available packets: 5
10. Kurkure - Solid Masti - price: 20, number of available packets: 5
11. Kurkure - Chataka Pataka - price: 20, number of available packets: 5
12. Kurkure - Chili Chataka - price: 20, number of available packets: 5
13. Nachos - Peri Peri - price: 30, number of available packets: 5
14. Nachos - Barbeque - price: 30, number of available packets: 5
15. Nachos - Sizzlin Jalapeno - price: 30, number of available packets: 5
16. Nachos - Tomato Mexican - price: 30, number of available packets: 5
17. Nachos - Tikka Masala - price: 30, number of available packets: 5
18. Biscuits - Parle-G - price: 20, number of available packets: 5
19. Biscuits - Bourbon - price: 30, number of available packets: 5
20. Biscuits - Monaco - price: 10, number of available packets: 5
21. Biscuits - Jim Jam - price: 40, number of available packets: 5
22. Biscuits - Good Day - price: 30, number of available packets: 5
23. Khakra - Methi - price: 40, number of available packets: 5
24. Khakra - Masala - price: 50, number of available packets: 5
25. Khakra -Plain - price: 60, number of available packets: 5
26. Khakra - Jeera - price: 30, number of available packets: 5
27. Khakra - Chili - price: 55, number of available packets: 5
28. Chocolate - Dairy Milk - price: 10, number of available packets: 5
```

```
5. Lays - Chile Limon - price: 20, number of available packets: 5
6. Lays - Classic Salted - price: 40, number of available packets: 5
7. Kurkure - Masala Munch - price: 20, number of available packets: 5
8. Kurkure - Puff Corn - price: 20, number of available packets: 5
9. Kurkure - Triangles - price: 20, number of available packets: 5
10. Kurkure - Solid Masti - price: 20, number of available packets: 5
11. Kurkure - Chataka Pataka - price: 20, number of available packets: 5
12. Kurkure - Chili Chataka - price: 20, number of available packets: 5
13. Nachos - Peri Peri - price: 30, number of available packets: 5
14. Nachos - Barbeque - price: 30, number of available packets: 5
15. Nachos - Sizzlin Jalapeno - price: 30, number of available packets: 5
16. Nachos - Tomato Mexican - price: 30, number of available packets: 5
17. Nachos - Tikka Masala - price: 30, number of available packets: 5
18. Biscuits - Parle-G - price: 20, number of available packets: 5
19. Biscuits - Bourbon - price: 30, number of available packets: 5
20. Biscuits - Monaco - price: 10, number of available packets: 5
21. Biscuits - Jim Jam - price: 40, number of available packets: 5
22. Biscuits - Good Day - price: 30, number of available packets: 5
23. Khakra - Methi - price: 40, number of available packets: 5
24. Khakra - Masala - price: 50, number of available packets: 5
25. Khakra -Plain - price: 60, number of available packets: 5
26. Khakra - Jeera - price: 30, number of available packets: 5
27. Khakra - Chili - price: 55, number of available packets: 5
28. Chocolate - Dairy Milk - price: 10, number of available packets: 5
29. Chocolate - KitKat - price: 25, number of available packets: 5
30. Chocolate - Munch - price: 10, number of available packets: 5
31. Chocolate - 5Star - price: 20, number of available packets: 5
32. Chocolate - Amul Dark Chocolate - price: 80, number of available packets: 5
33. Drinks - Thumbs-up - price: 40, number of available packets: 5
34. Drinks - Fanta - price: 40, number of available packets: 5
35. Drinks - Sprite - price: 40, number of available packets: 5
36. Drinks - Coca-cola - price: 40, number of available packets: 5
37. Drinks - Limca - price: 40, number of available packets: 5
38. Namkeen - Bhakharvadi - price: 65, number of available packets: 5
39. Namkeen - Aloo bhujiya - price: 35, number of available packets: 5
40. Namkeen - Lite mixture - price: 35, number of available packets: 5
41. Namkeen - Salted peanuts - price: 50, number of available packets: 5
42. Namkeen - Kachori - price: 60, number of available packets: 5
Are you ready to order - if not - enter 1 else enter any integer: 2

Enter item number: 3
enter quantity of packets: 2
please pay 50 rupees and collect your purchase(s): 2 Lays - American Style Cream And Onion.

Do you want to purchase more? if yes, enter 1 else enter any integer: 3

Thank You for shopping with us!
Good food brings good health :)
```

```
Are you a:
1.Customer
2.Manager
3.Or do you want to Leave application

enter choice: 1

choose one of the following options to see available products:
1.whole menu
2.categorically filter
3.filter on the basis of taste
4.filter on the basis of the price range
5.directly order
6.don't want to order

enter choice: 5

Enter item number: 16
enter quantity of packets: 7
we don't have that much quantity, do you want to enter a smaller value? (press 1 if yes, else press any integer): 1
enter quantity of packets: 4
please pay 120 rupees and collect your purchase(s): 4 Nachos - Tomato Mexican.

Do you want to purchase more? if yes, enter 1 else enter any integer: 7

Thank You for shopping with us!
Good food brings good health :)

Are you a:
1.Customer
2.Manager
3.Or do you want to Leave application

enter choice:
```

peoject.c - Untitled (Workspace) - Visual Studio Code

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
1.whole menu
2.categorically filter
3.filter on the basis of taste
4.filter on the basis of the price range
5.directly order
6.don't want to order

enter choice: 2

1-Lays
2-Kurkure
3-Chocolate
4-Drinks
5-Namkeen
6-Khakra
7-Nachos
8-Biscuits

If you want to know about the variety available then press 1(YES) else press 0(NO): 2

ERROR: please enter 1 or 0!!


If you want to know about the variety available then press 1(YES) else press 0(NO): 1

Please choose a number to know about a particular variety:2

7. Kurkure - Masala Munch - price: 20, number of available packets: 5
8. Kurkure - Puff Corn - price: 20, number of available packets: 5
9. Kurkure - Triangles - price: 20, number of available packets: 5
10. Kurkure - Solid Masti - price: 20, number of available packets: 5
11. Kurkure - Chataka Pataka - price: 20, number of available packets: 5
12. Kurkure - Chili Chataka - price: 20, number of available packets: 5

If you want to know about the variety available then press 1(YES) else press 0(NO): 0

We hope that was informative!

Are you ready to order - if not - enter 1 else enter any integer: 7

Enter item number: 10
enter quantity of packets: 6
we don't have that much quantity, do you want to enter a smaller value? (press 1 if yes, else press any integer): 5

Do you want to purchase more? if yes, enter 1 else enter any integer: 3

Thank You for shopping with us!
Good food brings good health :)
```

```
Are you a:
1.Customer
2.Manager
3.Or do you want to Leave application

enter choice: 2

enter admin pin: 12624

Enter item number: 14
enter quantity of packets you are adding: 20
we don't have that much capacity, do you want to enter a smaller value? (press 1 if yes, else press any integer): 10

Do you want to add more packets? if yes, enter 1 else enter any integer: 1

Enter item number: 16
enter quantity of packets you are adding: 5

Successfully added the packets!!
Updated number of packets of Nachos - Tomato Mexican: 6

Do you want to add more packets? if yes, enter 1 else enter any integer: 10

Are you a:
1.Customer
2.Manager
3.Or do you want to Leave application
```

```
Are you a:
1.Customer
2.Manager
3.Or do you want to Leave application

enter choice: 2

enter admin pin: 1234
invalid key!!
Do you want to enter again? press 1 if yes else any other integer: 1

enter admin pin: 12624

Enter item number: 3
enter quantity of packets you are adding: 5

Successfully added the packets!!
Updated number of packets of Lays - American Style Cream And Onion: 8

Do you want to add more packets? if yes, enter 1 else enter any integer: 3

Are you a:
1.Customer
2.Manager
3.Or do you want to Leave application

enter choice:
```

Project 3: Queue

LUDO

Code:

```
//task - create a program that uses queue with real-life applications
//LUDO

#include<stdio.h>
#include<stdlib.h>
#include<string.h>

struct player
{
   char name[60];
   int pos[4];
   int start;
   char s[6][3];
   int score;
   char home[4][3];
};

//Enqueue Dequeue
void enqueue(int a[4],int fr[2],int n)
{
   int front=fr[0];
   int rear=fr[1];
   if(front!=(rear+1)%4)
   {
      if (front==-1)
         fr[0]=0;
```

```
        fr[1]=(++rear)%4;
        a[fr[1]]=n;
    }
}

int dequeue(int a[4],int fr[2])
{
    int front=fr[0];
    int rear=fr[1];
    int n;
    if(front==rear==-1)
        return(0);
    n=a[front];
    if (front==rear)
    {
        fr[0]=-1;
        fr[1]=-1;
    }
    else
        fr[0]=(front+1)%4;
    return(n);
}

//displaying game board
void display(struct player p[],char a[52][3])
{
    printf("LUDO!!!\n\n");
    printf("
```

```
format of each piece numbering:\n");
```

```
   printf("       |      Player  1      | %s  | %s  | %s  |      Player  2      |
<player number><piece number>\n",a[10],a[11],a[12]);

   printf("       | ______   ______  |________|________|________|  ______   ______  |
<player number>* means 2 pieces\n");

   printf("       | |    | |    | | %s  | %s  | %s  | |    | |    | |                 <player
number># means 3 pieces\n",a[9],p[1].s[0],a[13]);

   printf("       | | %s  | | %s | |________|________|________| | %s  | | %s | |
<player number>$ means 4
pieces\n",p[0].home[0],p[0].home[1],p[1].home[0],p[1].home[1]);

   printf("       | |______|  |______| | %s  | %s  | %s  | |______|  |______|
|\n",a[8],p[1].s[1],a[14]);

   printf("       |                 |________|________|________|                 |
Player 1: %s\n",p[0].name);

   printf("       | ______   ______  | %s  | %s  | %s  | ______   ______  |
Player 2: %s\n",a[7],p[1].s[2],a[15],p[1].name);

   printf("       | |    | |    | |________|________|________| |    | |    | |
Player 3: %s\n",p[2].name);

   printf("       | | %s  | | %s | |  %s  |  %s  |  %s  | | %s  | | %s | |
Player 4:
%s\n",p[0].home[2],p[0].home[3],a[6],p[1].s[3],a[16],p[1].home[2],p[1].home[3],p[3].name);

   printf("       | |______|  |______| |________|________|________| |______|  |______|
|\n");

   printf("       |                 | %s  | %s  | %s  |                 |\n",a[5],p[1].s[4],a[17]);

   printf("
         |_____________________________|________|________|________|_____________________
___|\n");

   printf("       | | | | | | | |      | %s  |      | | | | | | |\n",p[1].s[5]);

   printf("       |%s |%s |%s |%s |%s |%s |      |________|       |%s |%s |%s |%s |%s |%s
|\n",a[51],a[0],a[1],a[2],a[3],a[4],a[18],a[19],a[20],a[21],a[22],a[23]);

   printf("       |___|___|___|___|___|___|____                ____|___|___|___|___|___|___|\n");

   printf("       | | | | | | | | |             | | | | | | | |\n");

   printf("       |%s |%s |%s |%s |%s |%s | %s |     LUDO      | %s |%s |%s |%s |%s |%s |%s
|\n",a[50],p[0].s[0],p[0].s[1],p[0].s[2],p[0].s[3],p[0].s[4],p[0].s[5],p[2].s[5],p[2].s[4],p[2].s[3],
p[2].s[2],p[2].s[1],p[2].s[0],a[24]);

   printf("       |___|___|___|___|___|___|____|            |____|___|___|___|___|___|___|\n");

   printf("       | | | | | | |       ________       | | | | | | |\n");
```

```
   printf("       |%s |%s |%s |%s |%s |%s |       |  %s  |       |%s |%s |%s |%s |%s |%s
|\n",a[49],a[48],a[47],a[46],a[45],a[44],p[3].s[5],a[30],a[29],a[28],a[27],a[26],a[25]);

   printf("
         |___|___|___|___|___|___|_________|_________|_________|___|___|___|___|___|___|\n"
);

   printf("       |      Player  4      |  %s  |  %s  |  %s  |      Player  3
|\n",a[43],p[3].s[4],a[31]);

   printf("       |  _____    _____   |_________|_________|_________|  _____    _____
|\n");

   printf("       | |     |  |     | |  %s  |  %s  |  %s  | |     |  |     | |\n",a[42],p[3].s[3],a[32]);

   printf("       | | %s  |  | %s  | |_________|_________|_________| | %s  |  | %s  |
|\n",p[3].home[0],p[3].home[1],p[2].home[0],p[2].home[1]);

   printf("       | |_____|  |_____| |  %s  |  %s  |  %s  | |_____|  |_____|
|\n",a[41],p[3].s[2],a[33]);

   printf("       |                 |_________|_________|_________|                 |\n");

   printf("       |  _____    _____  |  %s  |  %s  |  %s  |  _____    _____
|\n",a[40],p[3].s[1],a[34]);

   printf("       | |     |  |     | |_________|_________|_________| |     |  |     | |\n");

   printf("       | | %s  |  | %s  | |  %s  |  %s  |  %s  | | %s  |  | %s  |
|\n",p[3].home[2],p[3].home[3],a[39],p[3].s[0],a[35],p[2].home[2],p[2].home[3]);

   printf("       | |_____|  |_____| |_________|_________|_________| |_____|  |_____|
|\n");

   printf("       |                 |  %s  |  %s  |  %s  |                 |\n",a[38],a[37],a[36]);

   printf("
         |_____________________________|_________|_________|_________|_____________________
___|\n");

   printf("\n");

}


void main()

{

   //player structure object for 4 players

   struct player p[4]={{.name="-",.pos={-1,-1,-1,-1},.start=-1,.s={"  ","  ","  ","  ","  ","  ","
"},.score=0,.home={"11","12","13","14"}},
```

```
    {.name="-",.pos={-1,-1,-1,-1},.start=12,.s={"  ","  ","  ","  ","
"},.score=0,.home={"21","22","23","24"}},

    {.name="-",.pos={-1,-1,-1,-1},.start=25,.s={"  ","  ","  ","  ","
"},.score=0,.home={"31","32","33","34"}},

    {.name="-",.pos={-1,-1,-1,-1},.start=38,.s={"  ","  ","  ","  ","
"},.score=0,.home={"41","42","43","44"}}};

    printf("LUDO!!!\n\n");

    //declaring variables

    int i,n,queue[4]={0,0,0,0},win=0,fr[2]={-1,-
1},cur_p,g,roll,piece,end[4]={0,0,0,0},efr[2]={-1,-1},pos,start;

    char element[3]="  ",a[52][3]={"  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","
","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","
","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  ","  "},name[60];

    //input for number of players

    do

    {

    printf("Enter the number of players(2 or 3 or 4): ");

    scanf("%d",&n);}while((n<2)||(n>4));

    //enqueue players

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

    {

        printf("Enter the name of player %d:",i+1);

        scanf("%s",name);

        strcpy(p[i].name,name);

        enqueue(queue,fr,i+1);

    }

    //game

    srand(time(NULL));

    while(win<n-1)

    {

        //dequeuing current player

        cur_p=dequeue(queue,fr);
```

```
element[0]=(char)(cur_p+48);
g=1;
//loop to ensure that if a player rolls a six or kills another players piece then they get another turn
while(g==1)
{
   printf("press any key to continue.");
   getch();
   system("cls");
   display(p,a);
   //dice roll
   printf("%s's turn, press any key to roll the dice\n",p[cur_p-1].name);
   getch();
   roll=(rand()%6)+1;
   printf("you rolled a %d\n",roll);
   //if all pieces in home and roll is not 6
   if(((p[cur_p-1].pos[0]==-1)||(p[cur_p-1].pos[0]==57))&&((p[cur_p-1].pos[1]==-1)||(p[cur_p-1].pos[1]==57))&&((p[cur_p-1].pos[2]==-1)||(p[cur_p-1].pos[2]==57))&&((p[cur_p-1].pos[3]==-1)||(p[cur_p-1].pos[3]==57))&&(roll!=6))
   {
      printf("no move\n");
      break;
   }
   //input for which piece to move
   do{
   printf("which piece do you want to move(please enter 1/2/3/4 and the piece should not be\nin your home if you did not roll a six): ");
   scanf("%d",&piece);}while(((piece<1)||(piece>4))||((p[cur_p-1].pos[(piece-1)%4]==-1)&&(roll!=6)));
   //if roll is more than piece can move
   if(roll+p[cur_p-1].pos[piece-1]>57)
      break;
```

```
        //discontinuing loop if roll is not 6
        if(roll!=6)
           g=0;
        //assigning new position to piece according to it's current state and clearing old
position of that piece's number
        if(p[cur_p-1].pos[piece-1]==-1)
        {
           p[cur_p-1].pos[piece-1]=1;
           strcpy(p[cur_p-1].home[piece-1],"  ");
        }
        else
        {
           if(p[cur_p-1].pos[piece-1]<52)
           {
              if(a[(p[cur_p-1].pos[piece-1]+p[cur_p-1].start)%52][1]==(char)(piece+48))
                 strcpy(a[(p[cur_p-1].pos[piece-1]+p[cur_p-1].start)%52],"  ");
              else if(a[(p[cur_p-1].pos[piece-1]+p[cur_p-1].start)%52][1]=='*')
                 for(i=0;i<4;i++)
                 {
                    if((i!=piece-1)&&(p[cur_p-1].pos[piece-1]==p[cur_p-1].pos[i]))
                    {
                       a[(p[cur_p-1].pos[piece-1]+p[cur_p-1].start)%52][1]=(char)(i+49);
                       break;
                    }
                 }
              else if(a[(p[cur_p-1].pos[piece-1]+p[cur_p-1].start)%52][1]=='#')
                 a[(p[cur_p-1].pos[piece-1]+p[cur_p-1].start)%52][1]='*';
              else if(a[(p[cur_p-1].pos[piece-1]+p[cur_p-1].start)%52][1]=='$')
                 a[(p[cur_p-1].pos[piece-1]+p[cur_p-1].start)%52][1]='#';
           }
           else if(p[cur_p-1].pos[piece-1]<57)
```

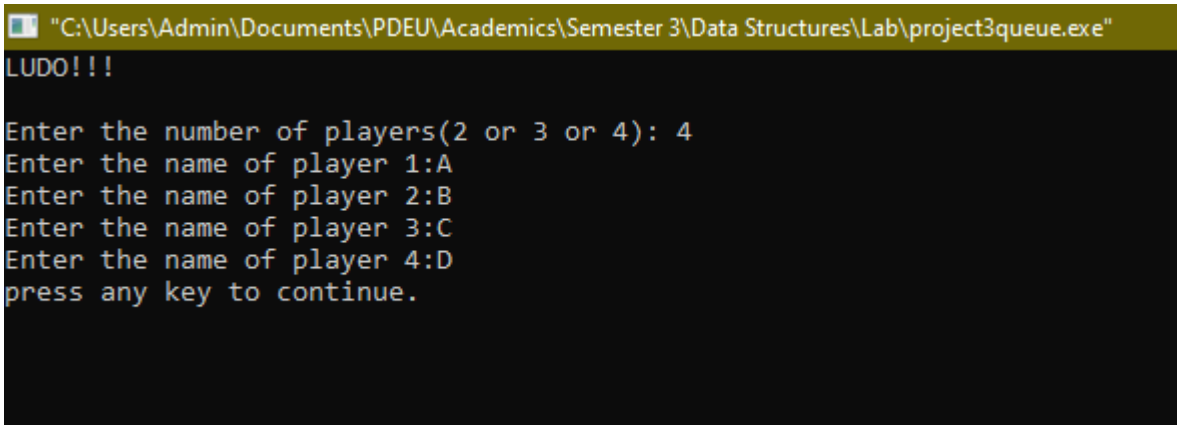
```
    {
        if(p[cur_p-1].s[p[cur_p-1].pos[piece-1]%52][1]==(char)(piece+48))
            strcpy(p[cur_p-1].s[p[cur_p-1].pos[piece-1]%52],"  ");
        else if(p[cur_p-1].s[p[cur_p-1].pos[piece-1]%52][1]=='*')
            for(i=0;i<4;i++)
            {
                if((i!=piece-1)&&(p[cur_p-1].pos[piece-1]==p[cur_p-1].pos[i]))
                {
                    p[cur_p-1].s[p[cur_p-1].pos[piece-1]%52][1]=(char)(i+49);
                    break;
                }
            }
        else if(p[cur_p-1].s[p[cur_p-1].pos[piece-1]%52][1]=='#')
            p[cur_p-1].s[p[cur_p-1].pos[piece-1]%52][1]='*';
        else if(p[cur_p-1].s[p[cur_p-1].pos[piece-1]%52][1]=='$')
            p[cur_p-1].s[p[cur_p-1].pos[piece-1]%52][1]='#';
    }
    p[cur_p-1].pos[piece-1]+=roll;
}
pos=p[cur_p-1].pos[piece-1];
element[1]=(char)(piece+48);
//updating special positions of particular player if the piece enters that queue
if(pos>=52)
{
    if(p[cur_p-1].s[pos%52][0]==' ')
        strcpy(p[cur_p-1].s[pos%52],element);
    else if(p[cur_p-1].s[pos%52][1]=='*')
        p[cur_p-1].s[pos%52][1]='#';
    else if(p[cur_p-1].s[pos%52][1]=='#')
        p[cur_p-1].s[pos%52][1]='$';
```

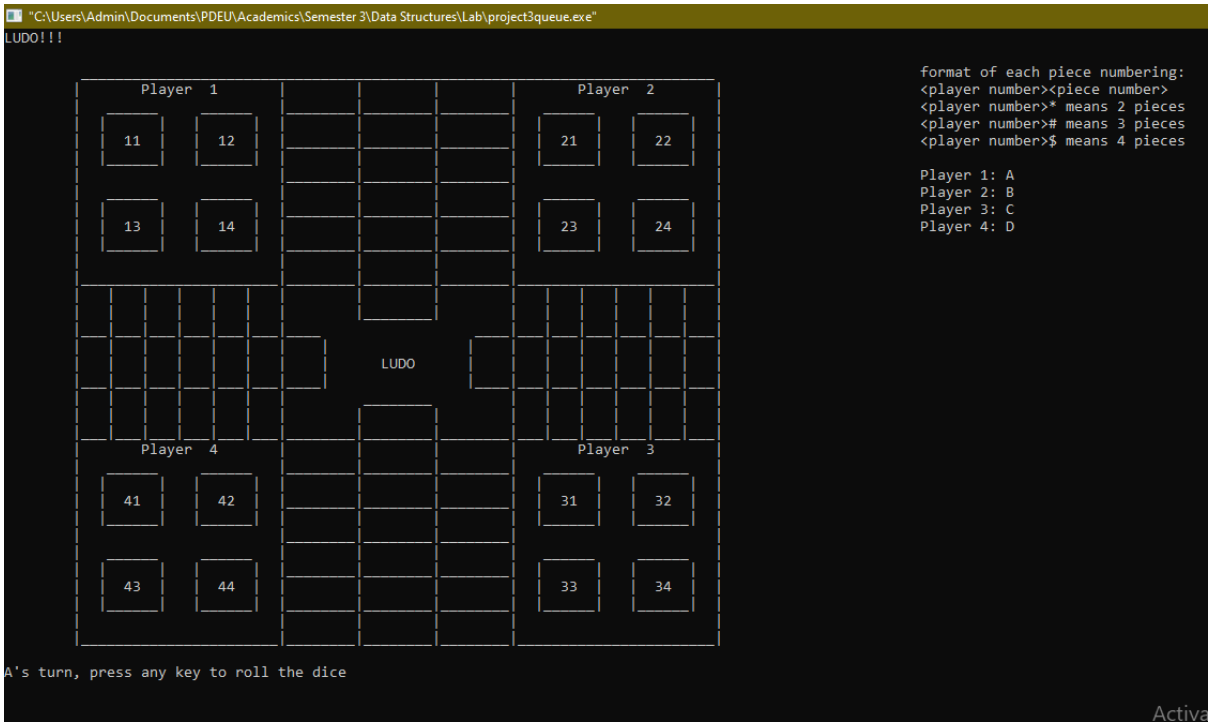
```
    else
        p[cur_p-1].s[pos%52][1]='*';
    if(pos==57)
    {
        p[cur_p-1].score+=1;
        if(p[cur_p-1].score==4)
            g=0;
    }
}
//updating the common positions queue
else
{
    start=p[cur_p-1].start;
    if(a[(pos+start)%52][0]==' ')
        strcpy(a[(pos+start)%52],element);
    //if a piece of same player is there in new place
    else if(a[(pos+start)%52][0]==element[0])
    {
        if(a[(pos+start)%52][1]=='*')
            a[(pos+start)%52][1]='#';
        else if(a[(pos+start)%52][1]=='#')
            a[(pos+start)%52][1]='$';
        else
            a[(pos+start)%52][1]='*';
    }
    //if a piece of another player is there in new place
    else
    {
        g=1;
        if(a[(pos+start)%52][1]=='*')
```

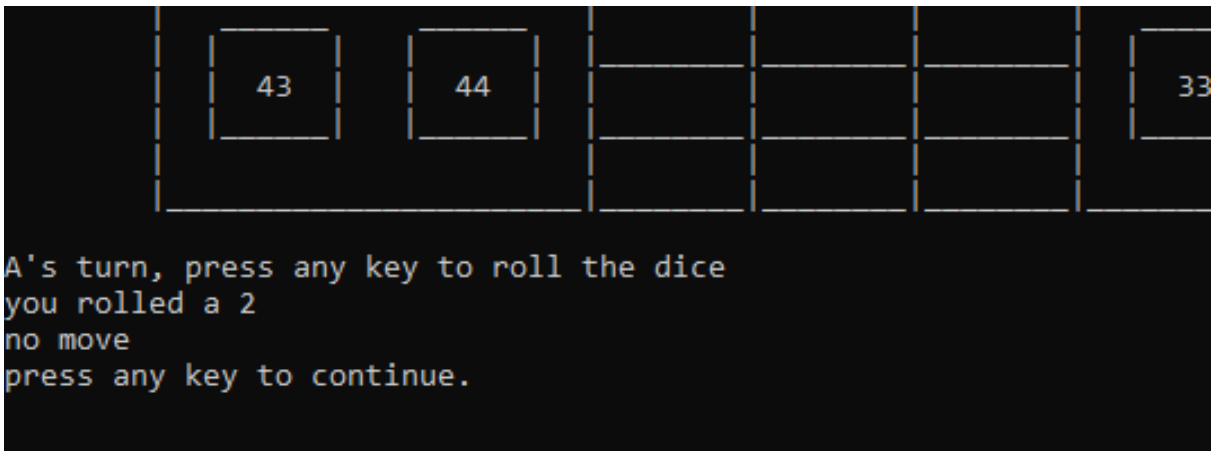
```
            {
               for(i=0;i<4;i++)
               {
                  if((pos+start)%52==(p[(int)a[(pos+start)%52][0]-
49].pos[i]+p[(int)a[(pos+start)%52][0]-49].start)%52)
                  {
                     p[(int)a[(pos+start)%52][0]-49].home[i][0]=a[(pos+start)%52][0];
                     p[(int)a[(pos+start)%52][0]-49].home[i][1]=(char)(i+49);
                     p[(int)a[(pos+start)%52][0]-49].pos[i]=-1;
                  }
               }
            }
            else if(a[(pos+start)%52][1]=='#')
            {
               for(i=0;i<4;i++)
               {
                  if((pos+start)%52==(p[(int)a[(pos+start)%52][0]-
49].pos[i]+p[(int)a[(pos+start)%52][0]-49].start)%52)
                  {
                     p[(int)a[(pos+start)%52][0]-49].home[i][0]=a[(pos+start)%52][0];
                     p[(int)a[(pos+start)%52][0]-49].home[i][1]=(char)(i+49);
                     p[(int)a[(pos+start)%52][0]-49].pos[i]=-1;
                  }
               }
            }
            else if(a[(pos+start)%52][1]=='$')
            {
               for(i=0;i<4;i++)
               {
                  p[(int)a[(pos+start)%52][0]-49].home[i][0]=a[(pos+start)%52][0];
                  p[(int)a[(pos+start)%52][0]-49].home[i][1]=(char)(i+49);
```

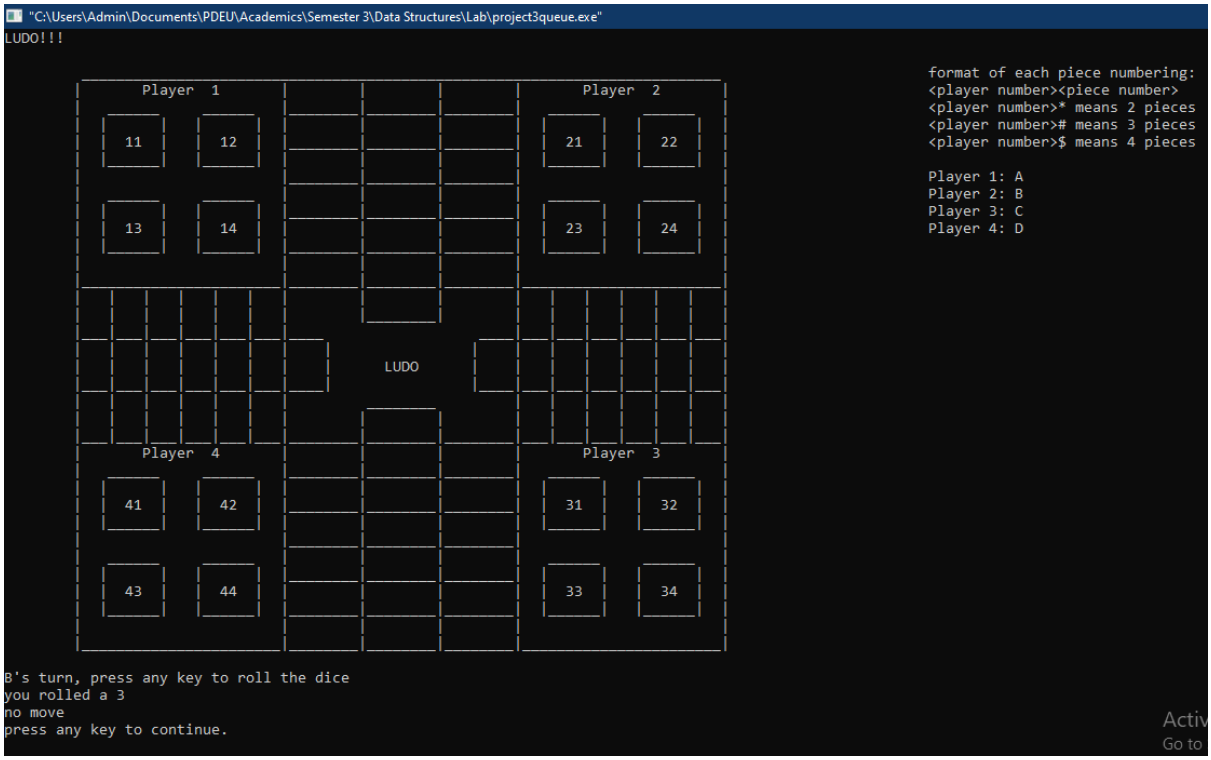
```
                    p[(int)a[(pos+start)%52][0]-49].pos[i]=-1;
                }
            }
            else
            {
                strcpy(p[(int)a[(pos+start)%52][0]-49].home[(int)a[(pos+start)%52][1]-49],a[(pos+start)%52]);
                p[(int)a[(pos+start)%52][0]-49].pos[(int)a[(pos+start)%52][1]-49]=-1;
            }
            strcpy(a[(pos+start)%52],element);
        }
    }
    }
    //updating end queue if player has pieces at end or else enqueueing the player
    if(p[cur_p-1].score==4)
    {
        win++;
        enqueue(end,efr,cur_p);
    }
    else
        enqueue(queue,fr,cur_p);
}
//end of game.
printf("Winners:\n");
for(i=0;i<n-1;i++)
    printf("%d: %s\n",i+1,p[end[i]-1].name);
}
```

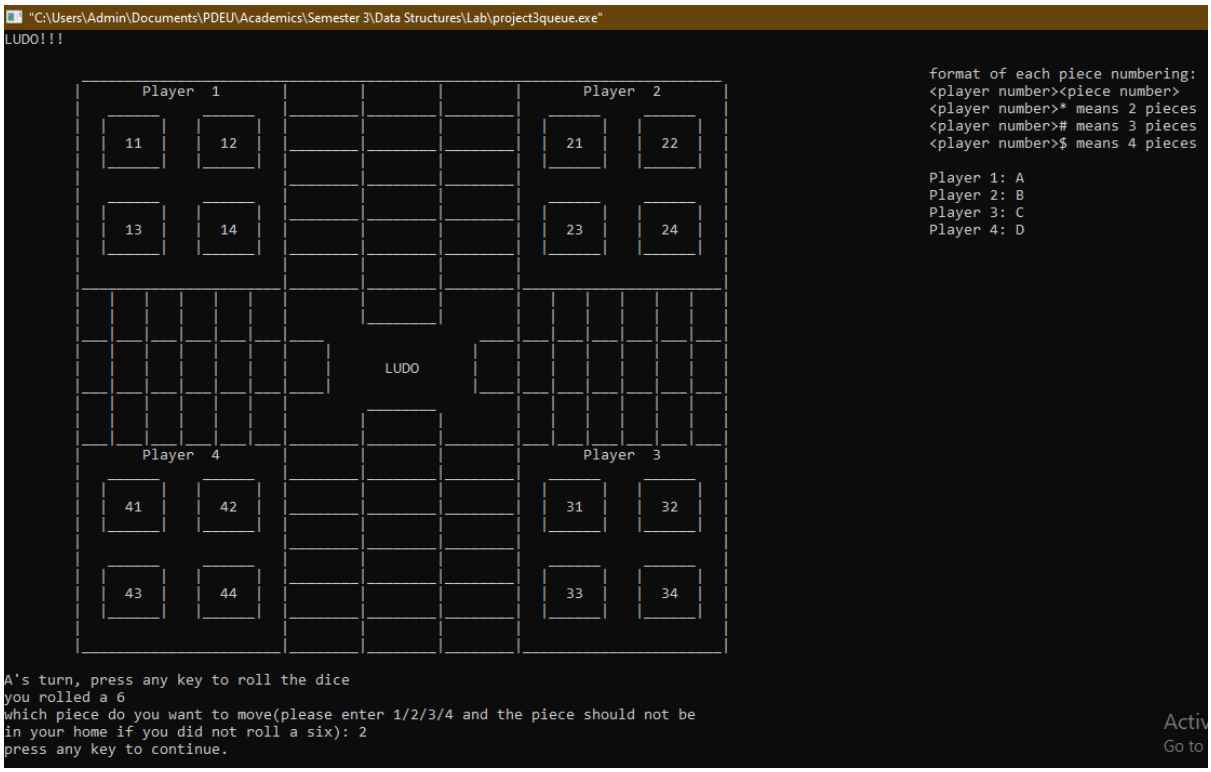
Output:

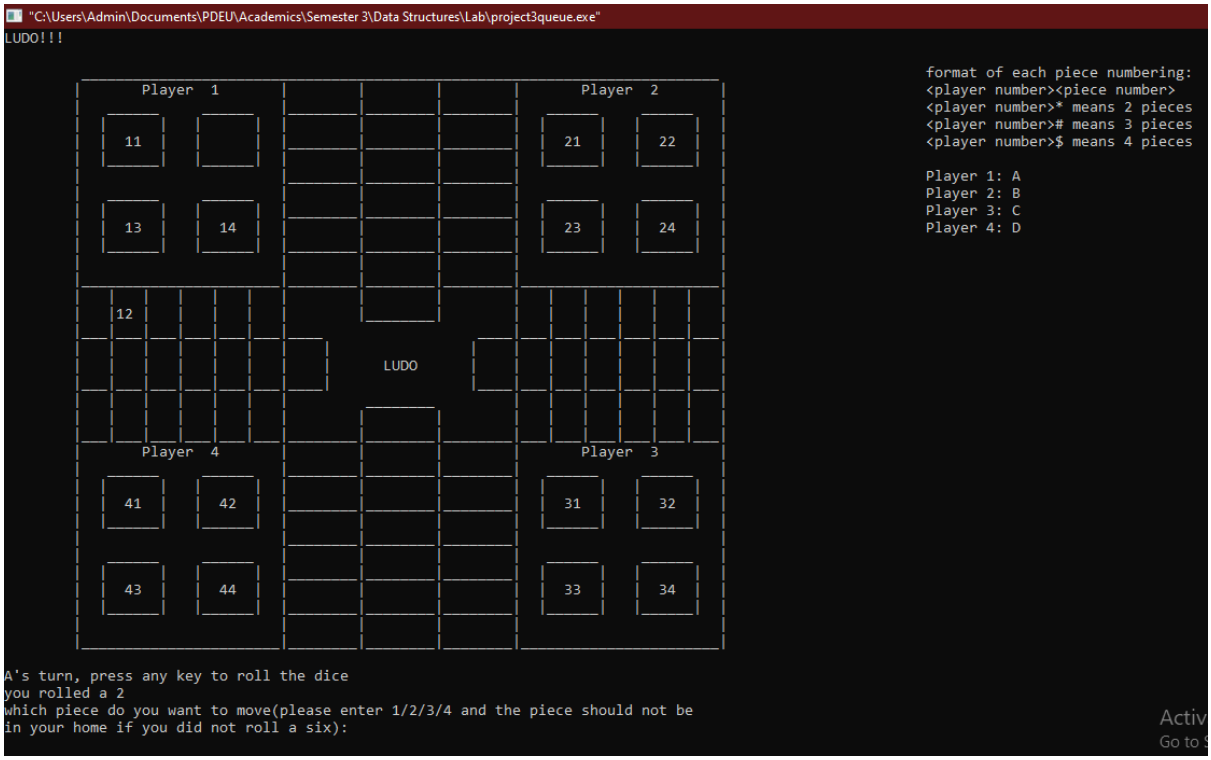

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Enter the number of players(2 or 3 or 4): 4
Enter the name of player 1:A
Enter the name of player 2:B
Enter the name of player 3:C
Enter the name of player 4:D
press any key to continue.

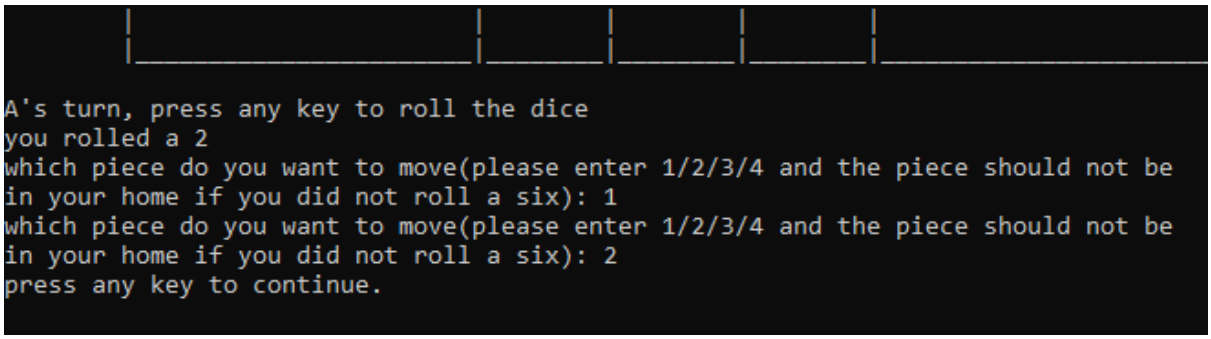

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 4
Player 3
LUDO
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
A's turn, press any key to roll the dice

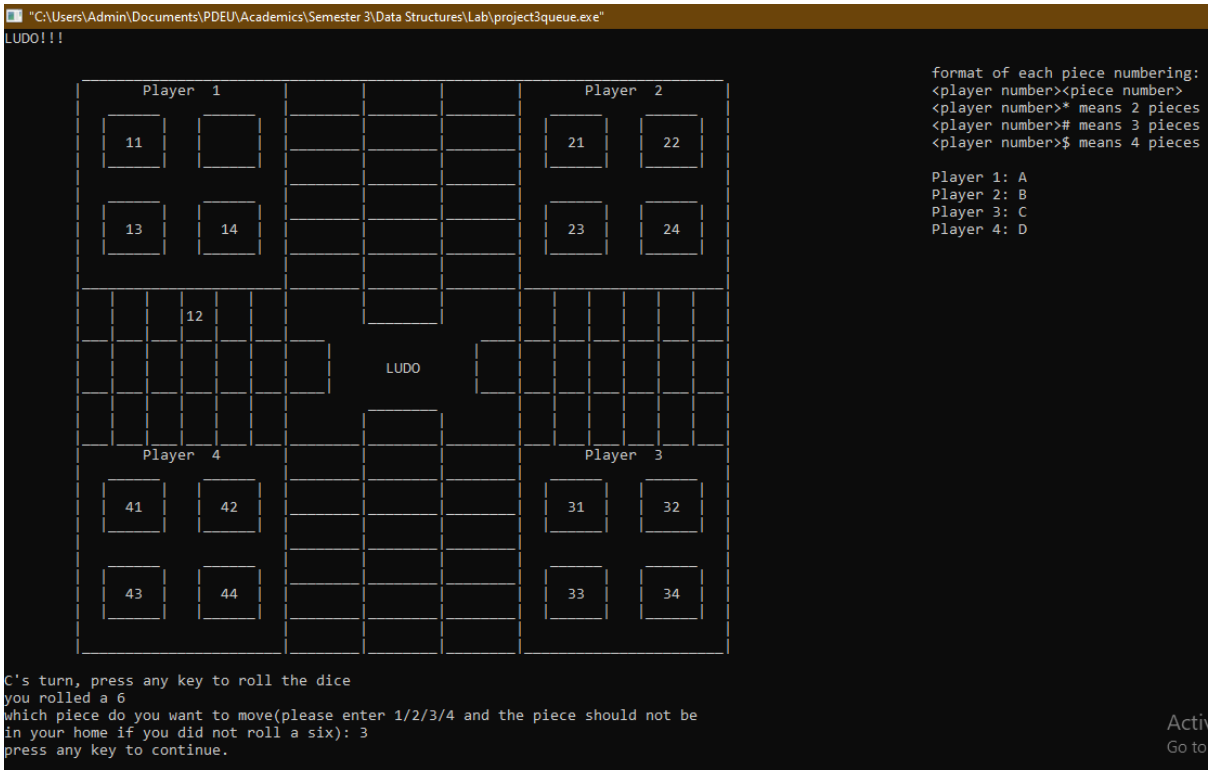

43
44
33
A's turn, press any key to roll the dice
you rolled a 2
no move
press any key to continue.

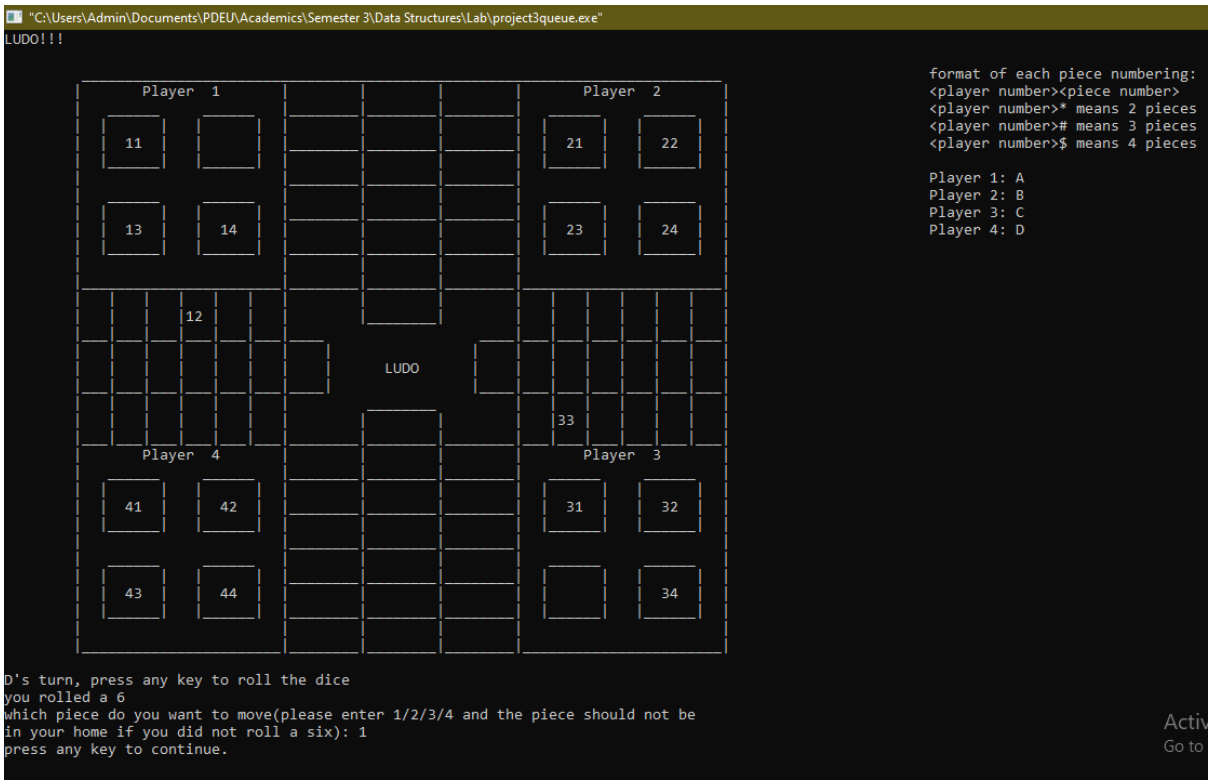

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 4
Player 3
11
12
13
14
21
22
23
24
41
42
43
44
31
32
33
34
LUDO
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
B's turn, press any key to roll the dice
you rolled a 3
no move
press any key to continue.

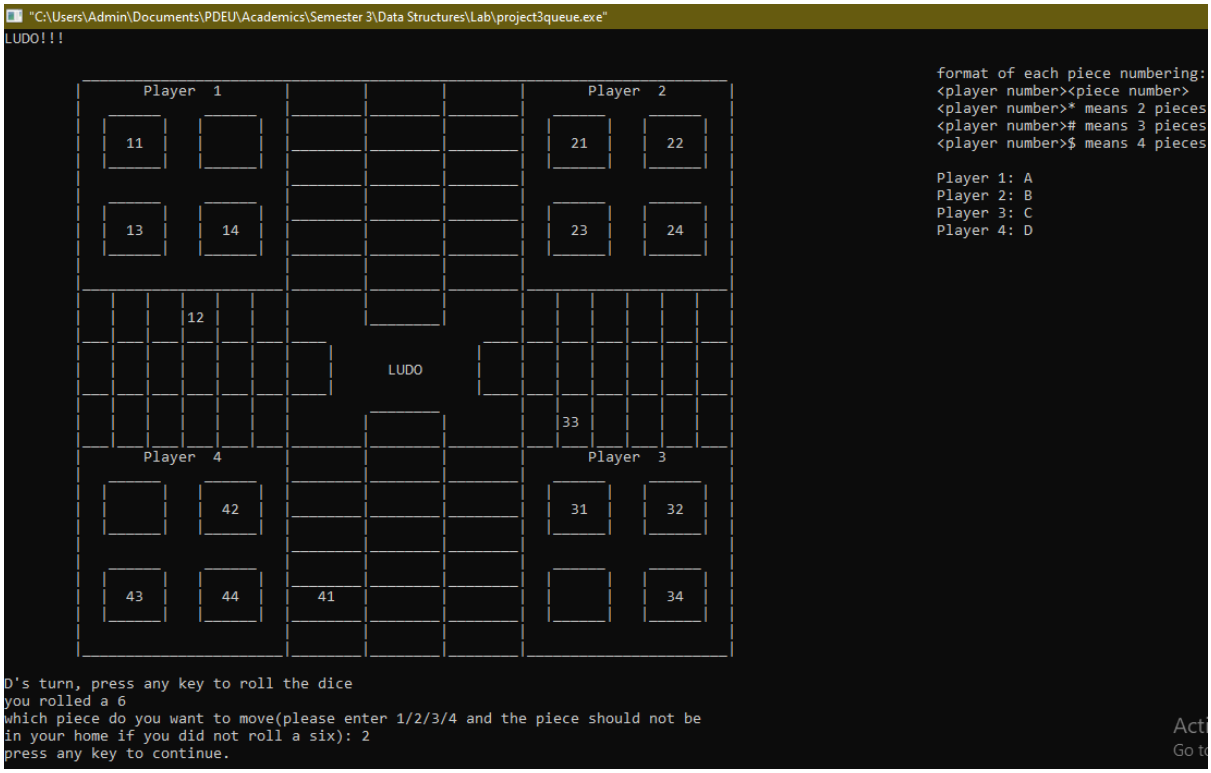

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 4
Player 3
11
12
13
14
21
22
23
24
41
42
43
44
31
32
33
34
LUDO
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
A's turn, press any key to roll the dice
you rolled a 6
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six): 2
press any key to continue.

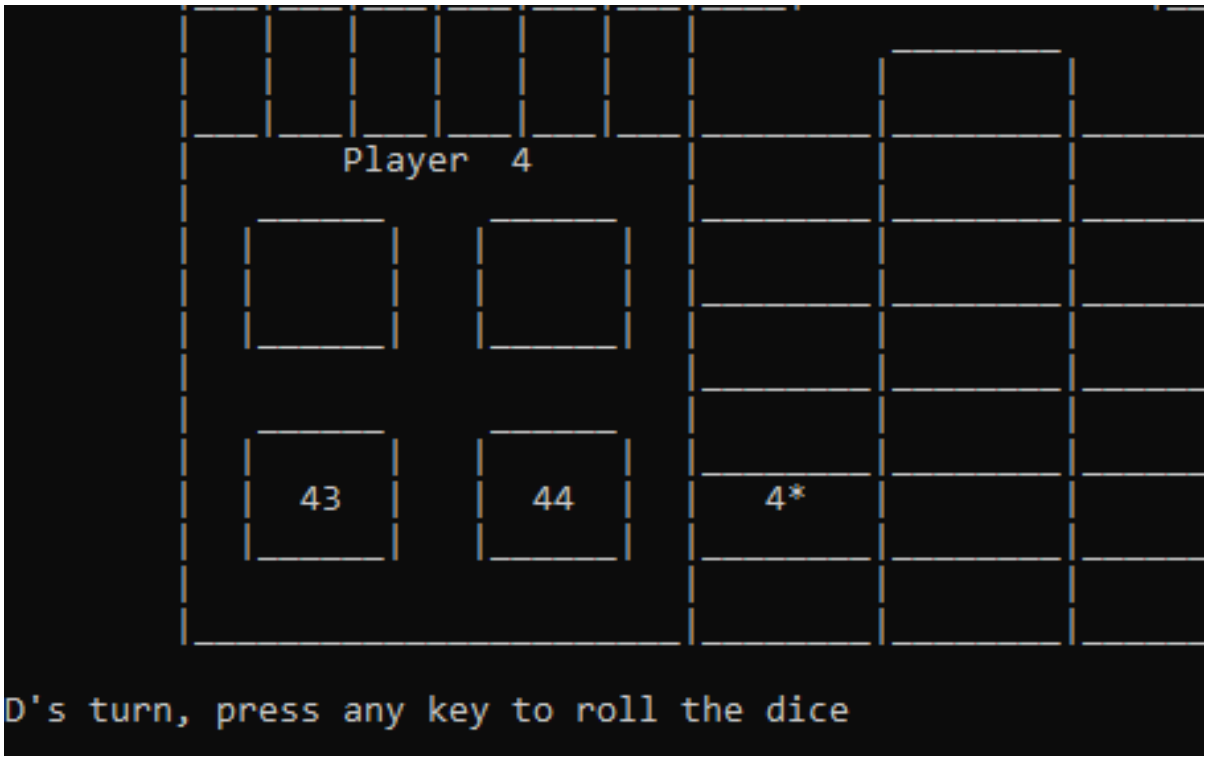

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 4
Player 3
LUDO
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
A's turn, press any key to roll the dice
you rolled a 2
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six):

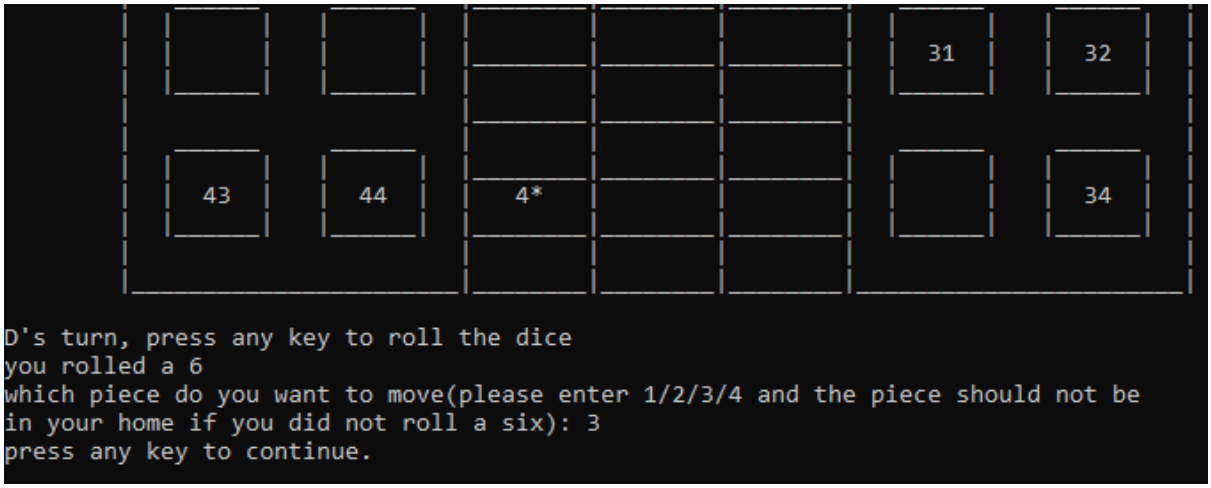

A's turn, press any key to roll the dice
you rolled a 2
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six): 1
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six): 2
press any key to continue.

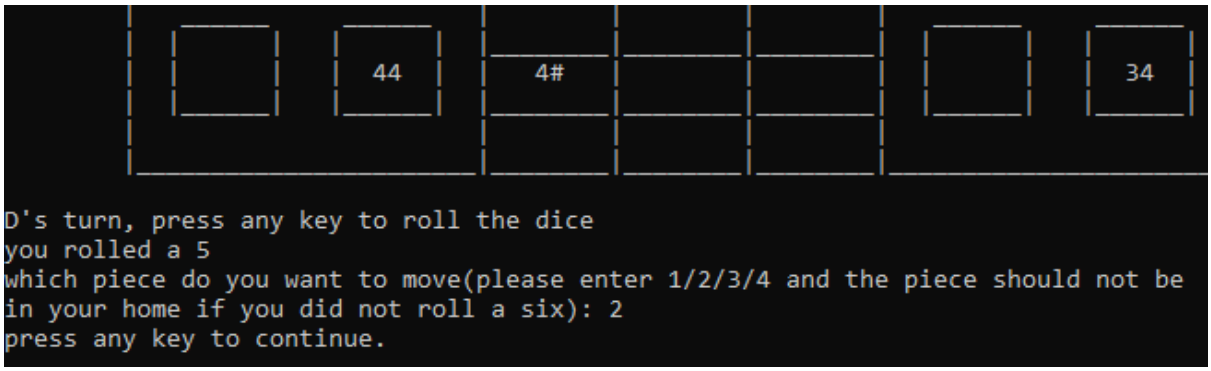

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 4
Player 3
11
12
13
14
21
22
23
24
41
42
43
44
31
32
33
34
LUDO
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
C's turn, press any key to roll the dice
you rolled a 6
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six): 3
press any key to continue.

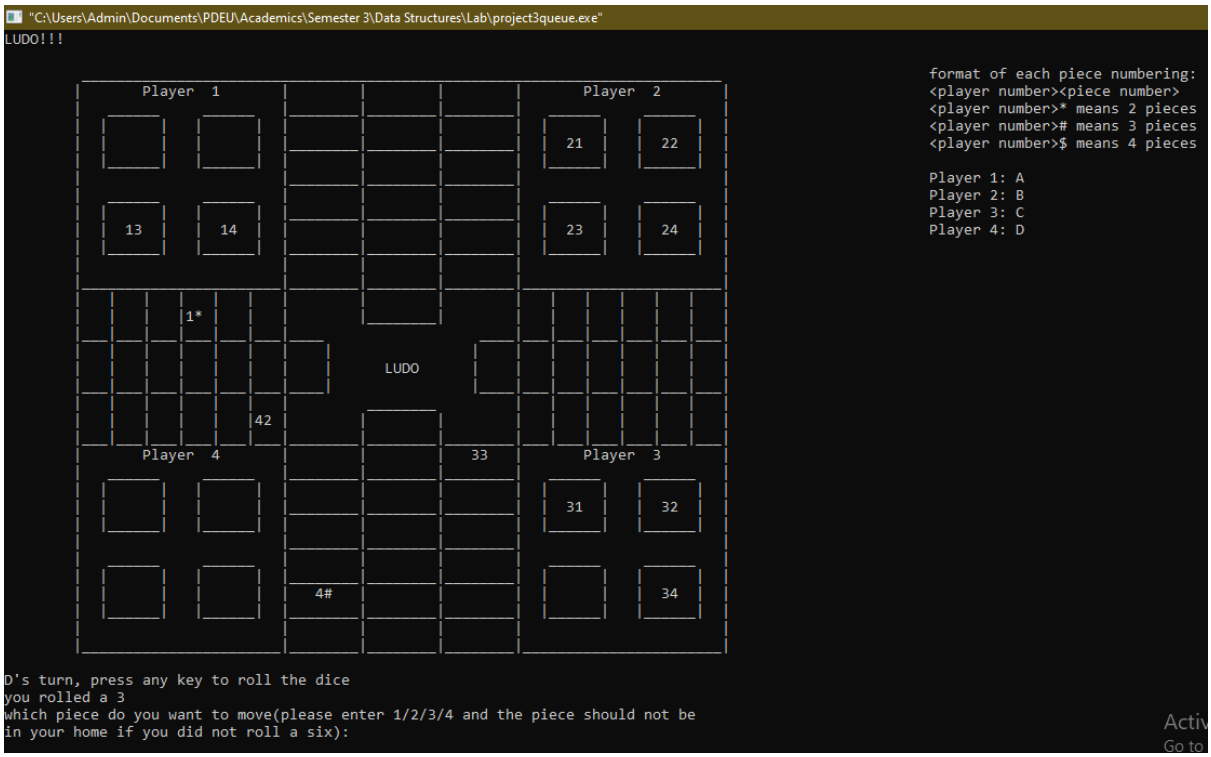

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 4
Player 3
11
12
13
14
21
22
23
24
41
42
43
44
31
32
33
34
LUDO
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
D's turn, press any key to roll the dice
you rolled a 6
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six): 1
press any key to continue.

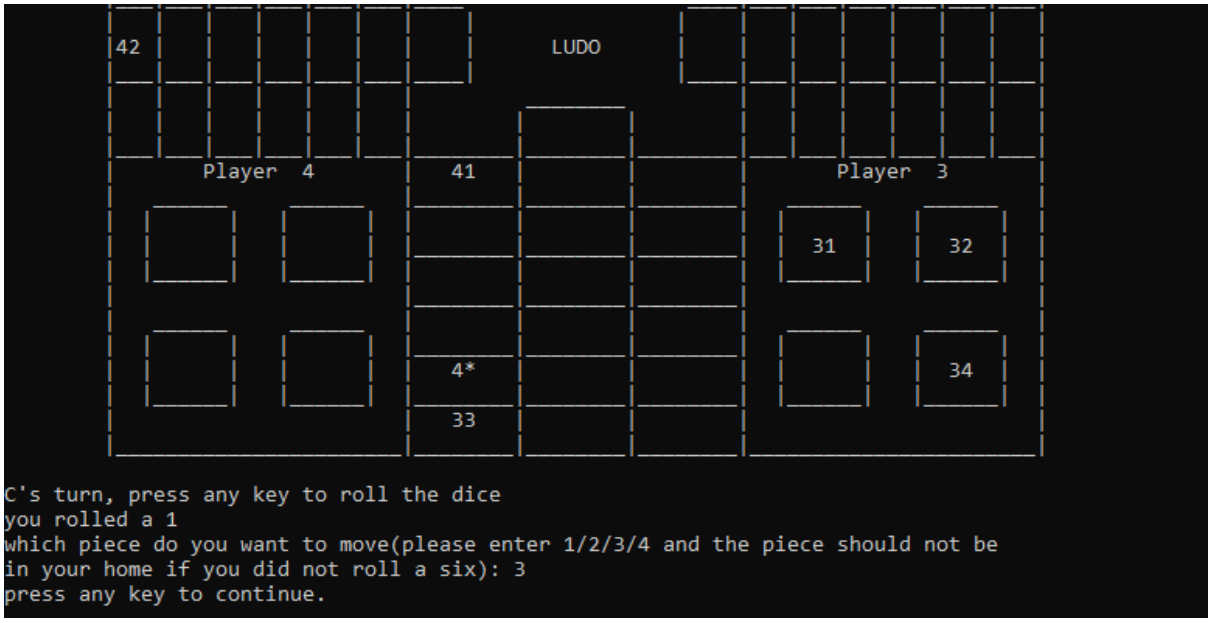

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 4
Player 3
LUDO
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
D's turn, press any key to roll the dice
you rolled a 6
Which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six): 2
press any key to continue.

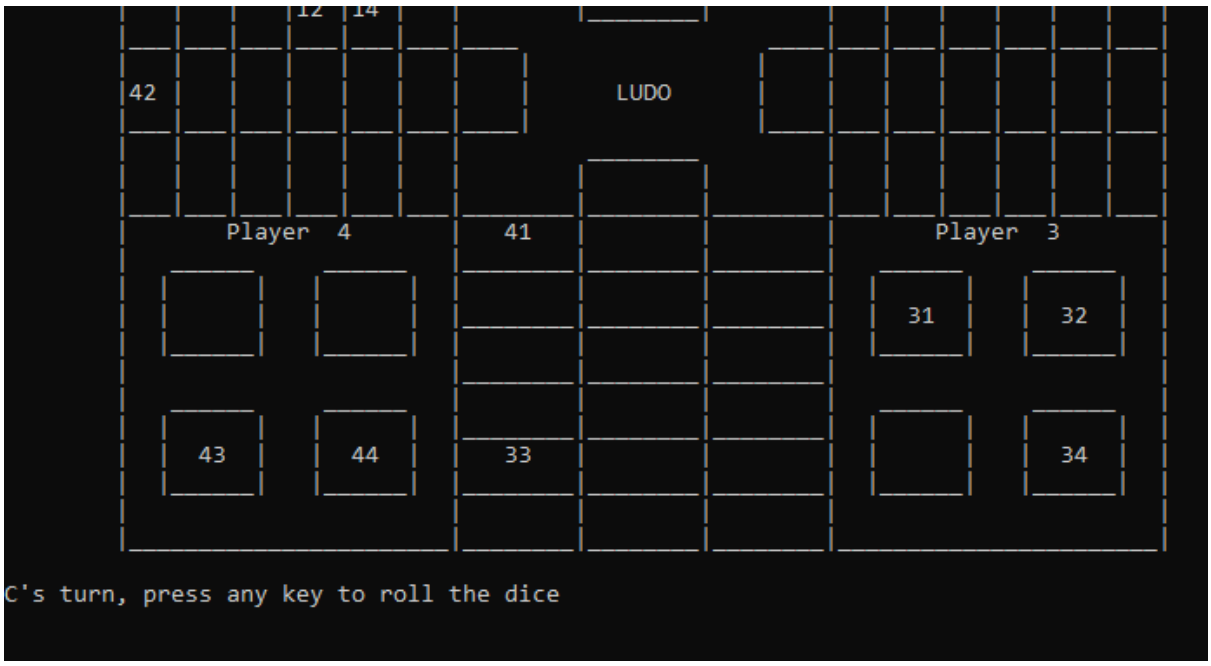

Player 4
43
44
4*
D's turn, press any key to roll the dice

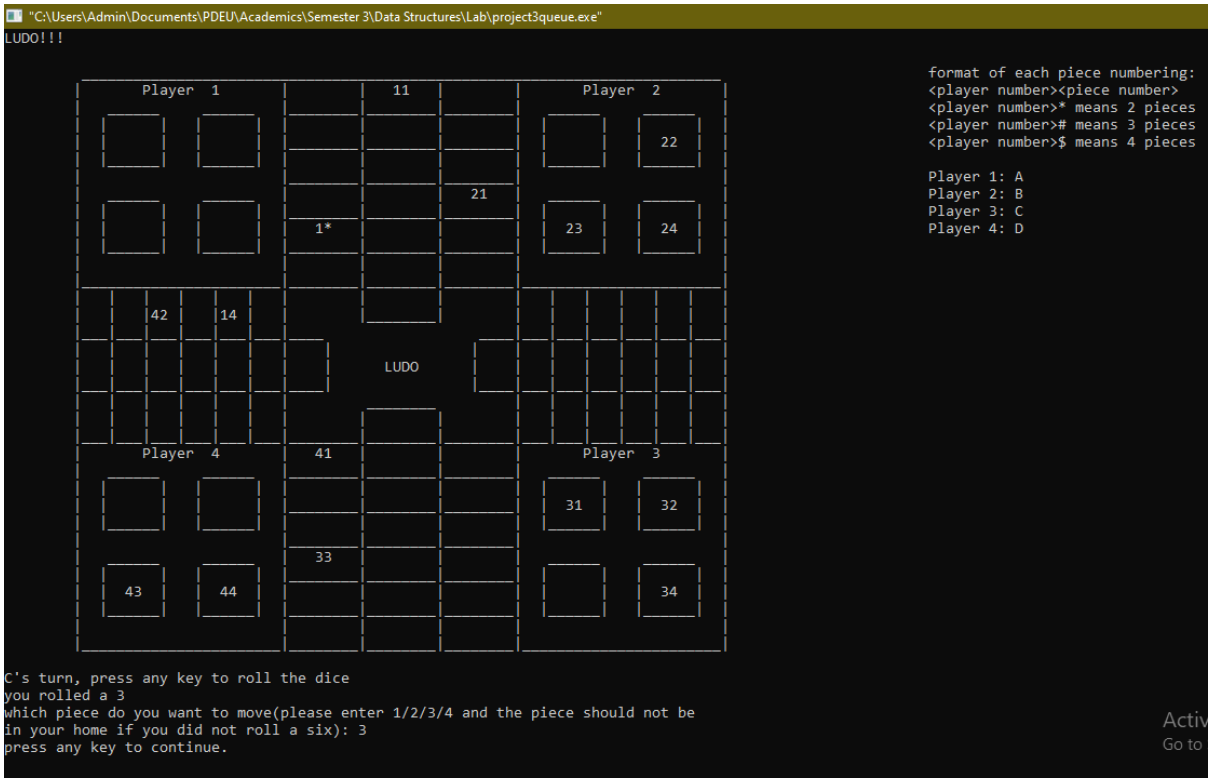

31
32
43
44
4*
34
D's turn, press any key to roll the dice
you rolled a 6
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six): 3
press any key to continue.

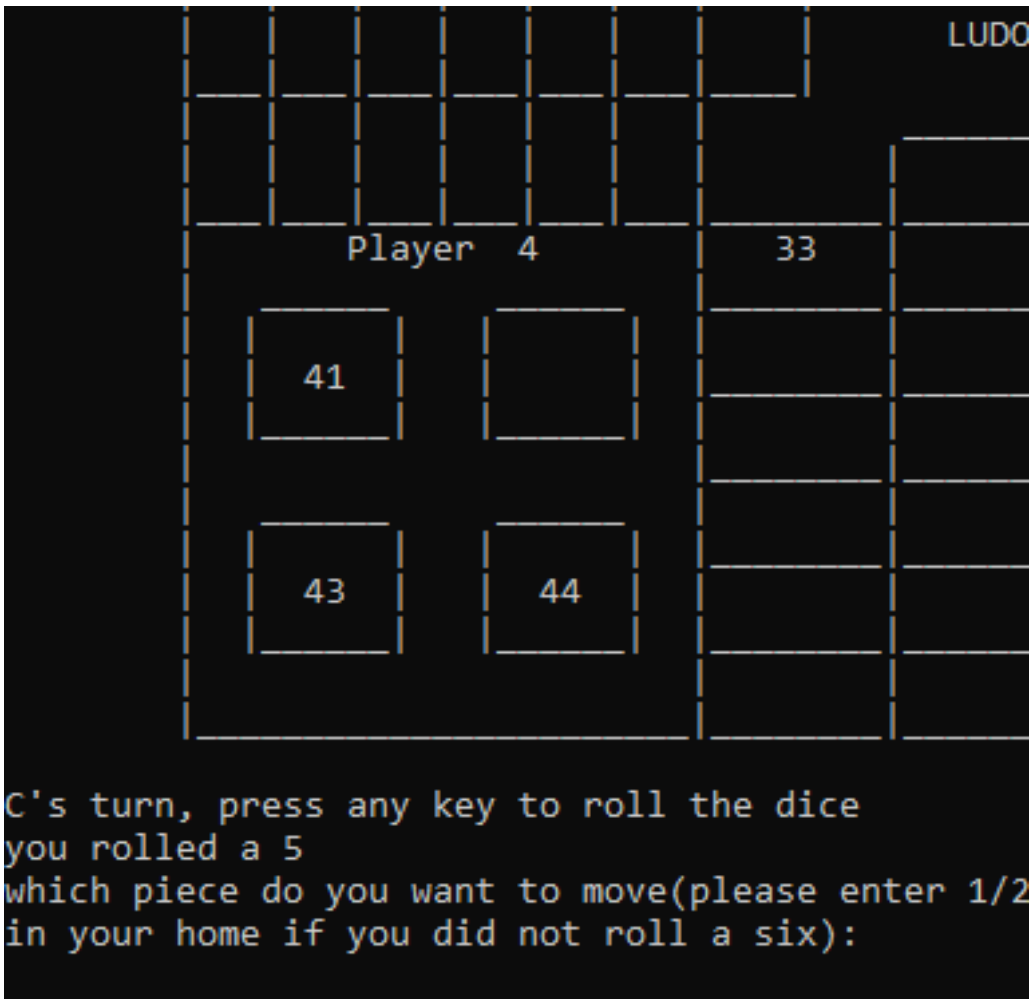

44
4#
34
D's turn, press any key to roll the dice
you rolled a 5
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six): 2
press any key to continue.

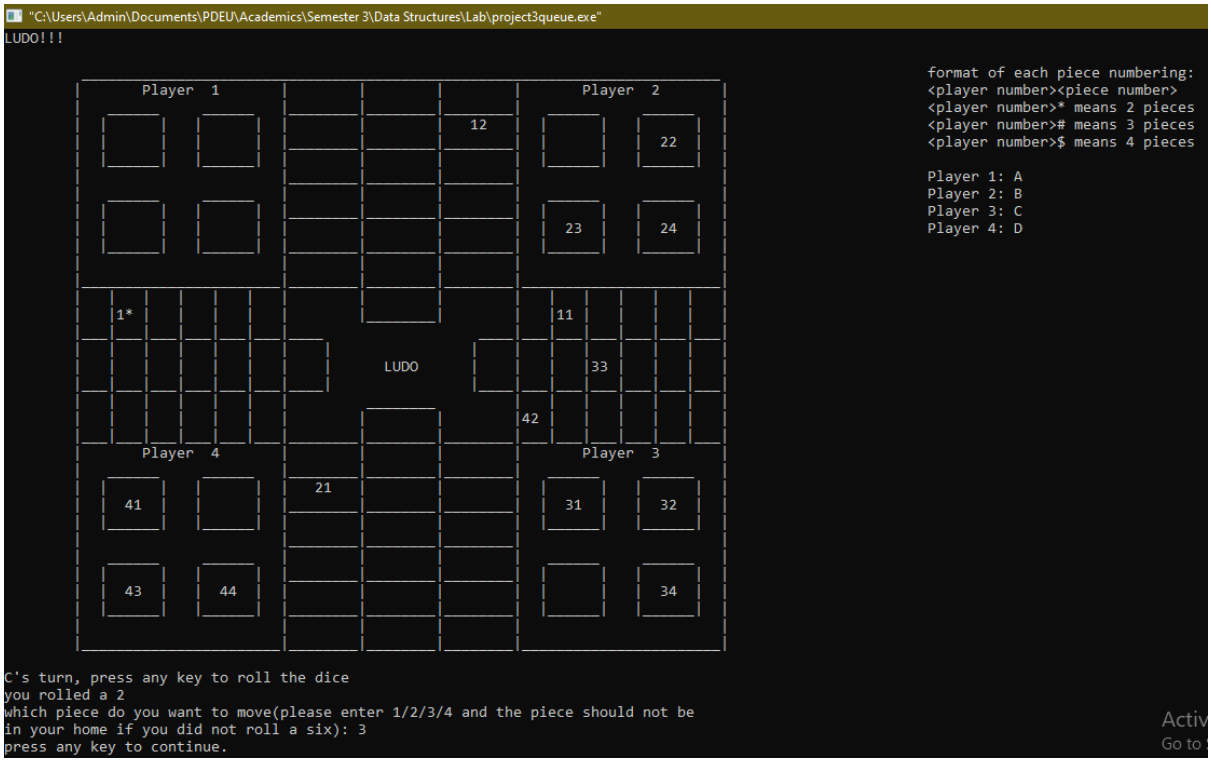

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 4
Player 3
LUDO
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
D's turn, press any key to roll the dice
you rolled a 3
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six):

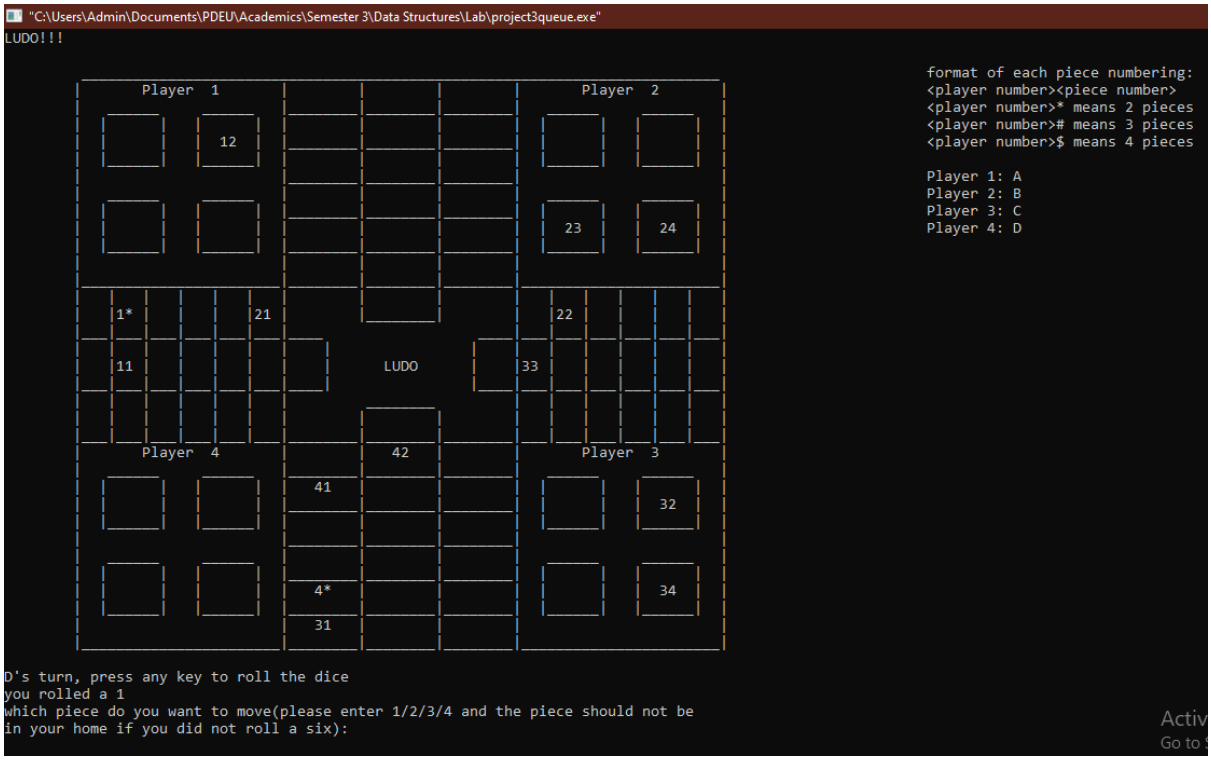

42
LUDO
Player 4
41
Player 3
31
32
4*
34
33
C's turn, press any key to roll the dice
you rolled a 1
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six): 3
press any key to continue.

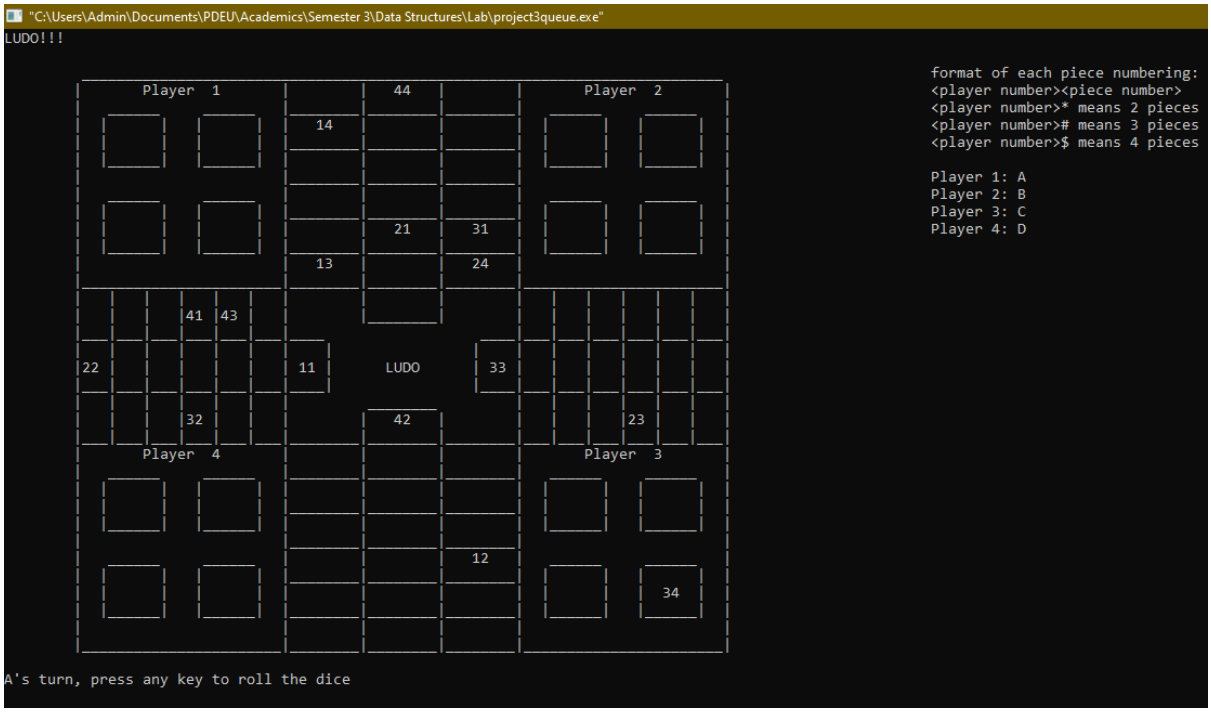

12
14
42
LUDO
Player 4
41
Player 3
31
32
43
44
33
34
C's turn, press any key to roll the dice

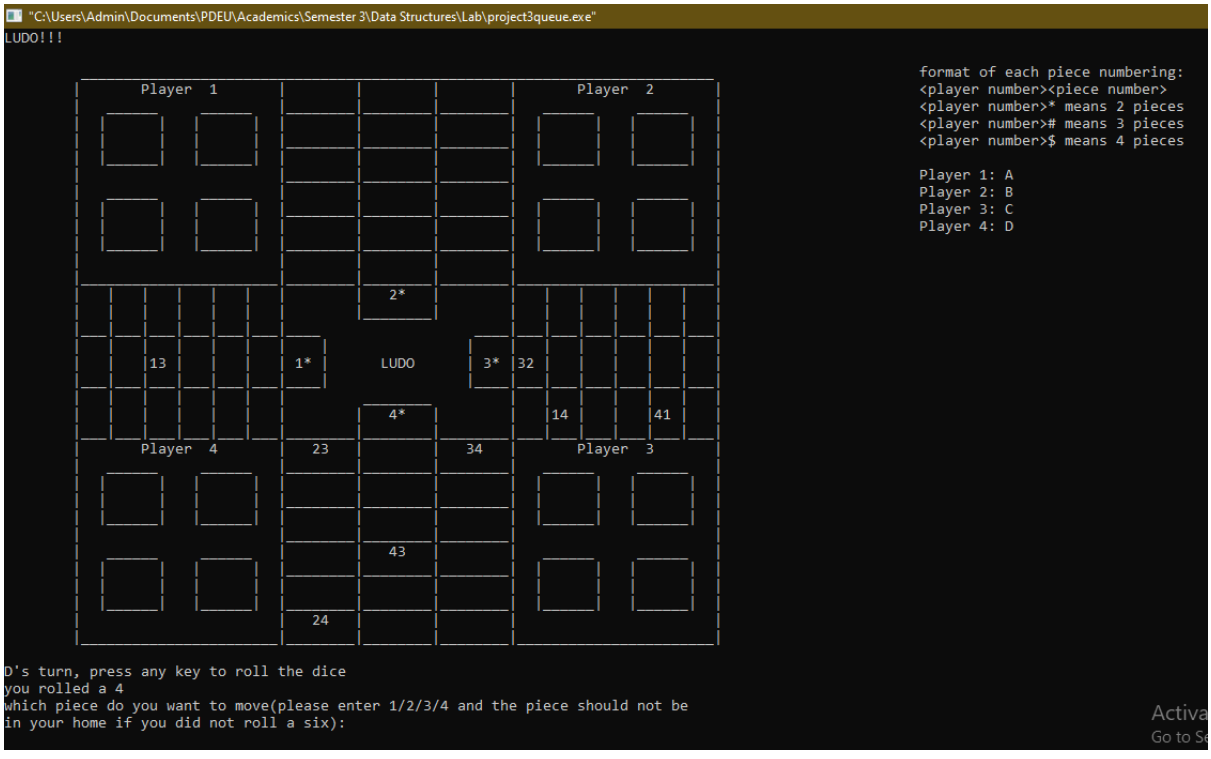

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 4
Player 3
LUDO
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
C's turn, press any key to roll the dice
you rolled a 3
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six): 3
press any key to continue.

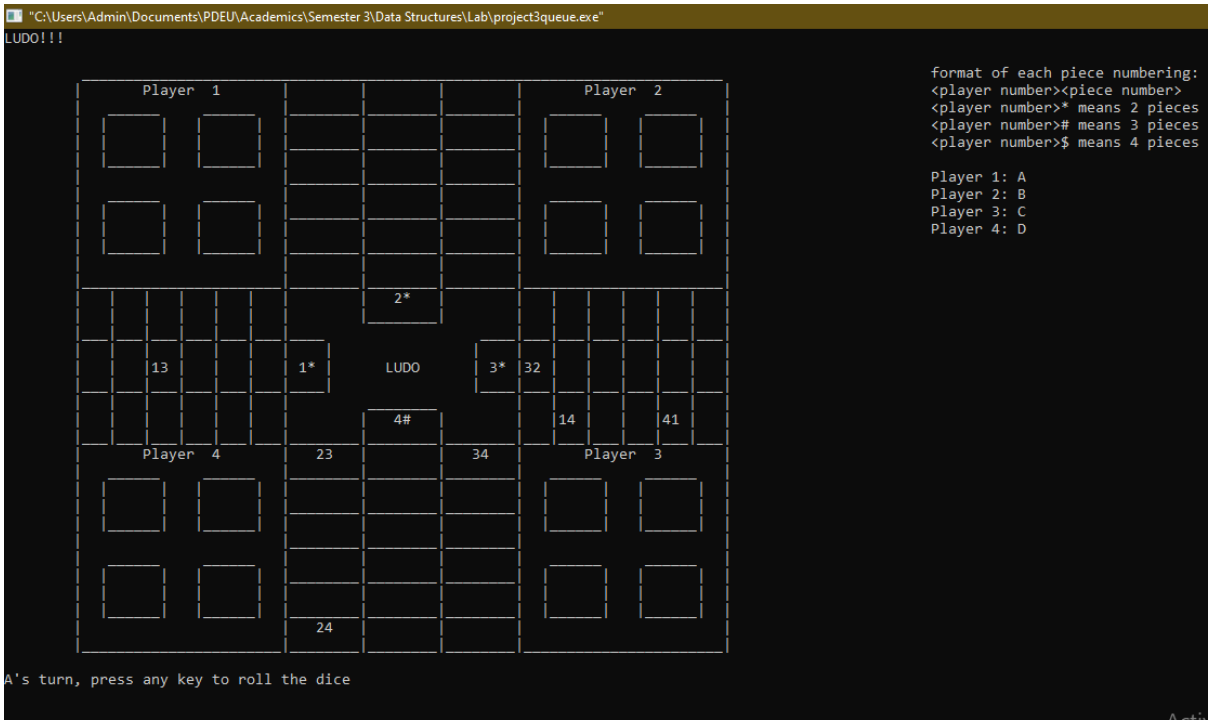

LUDO
Player 4
33
41
43
44
C's turn, press any key to roll the dice
you rolled a 5
which piece do you want to move(please enter 1/2
in your home if you did not roll a six):

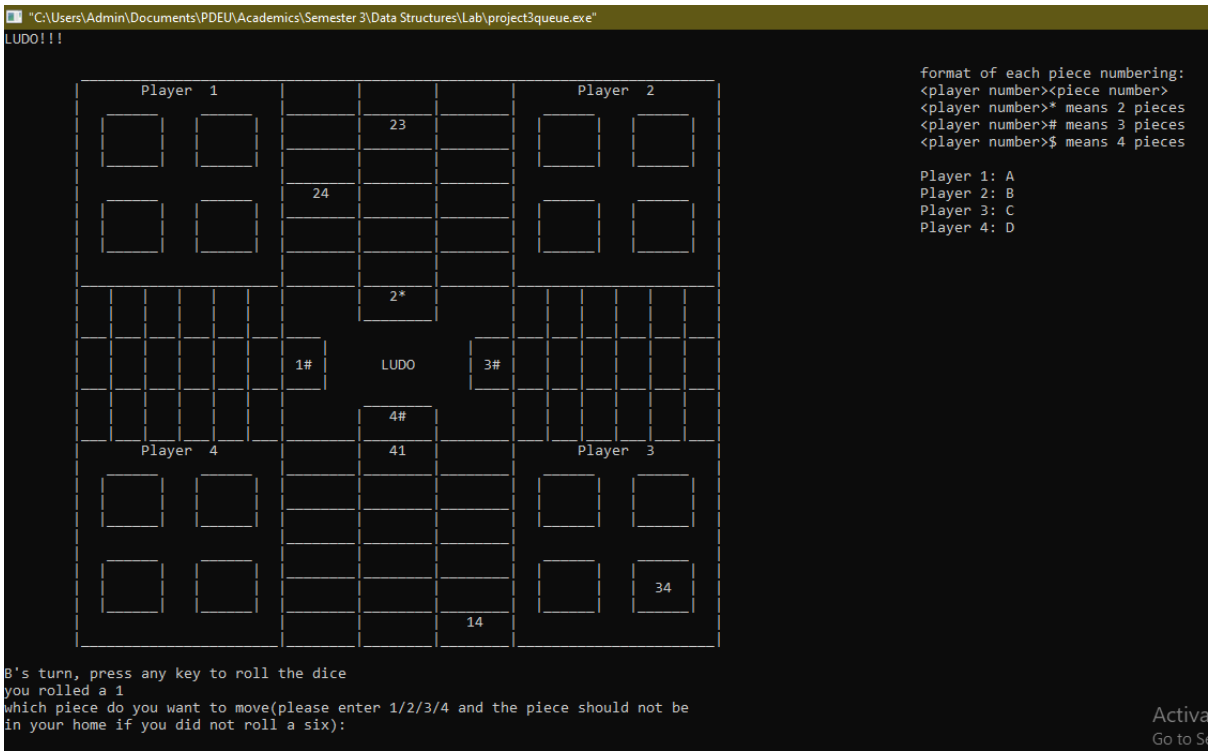

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 3
Player 4
LUDO
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
C's turn, press any key to roll the dice
you rolled a 2
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six): 3
press any key to continue.

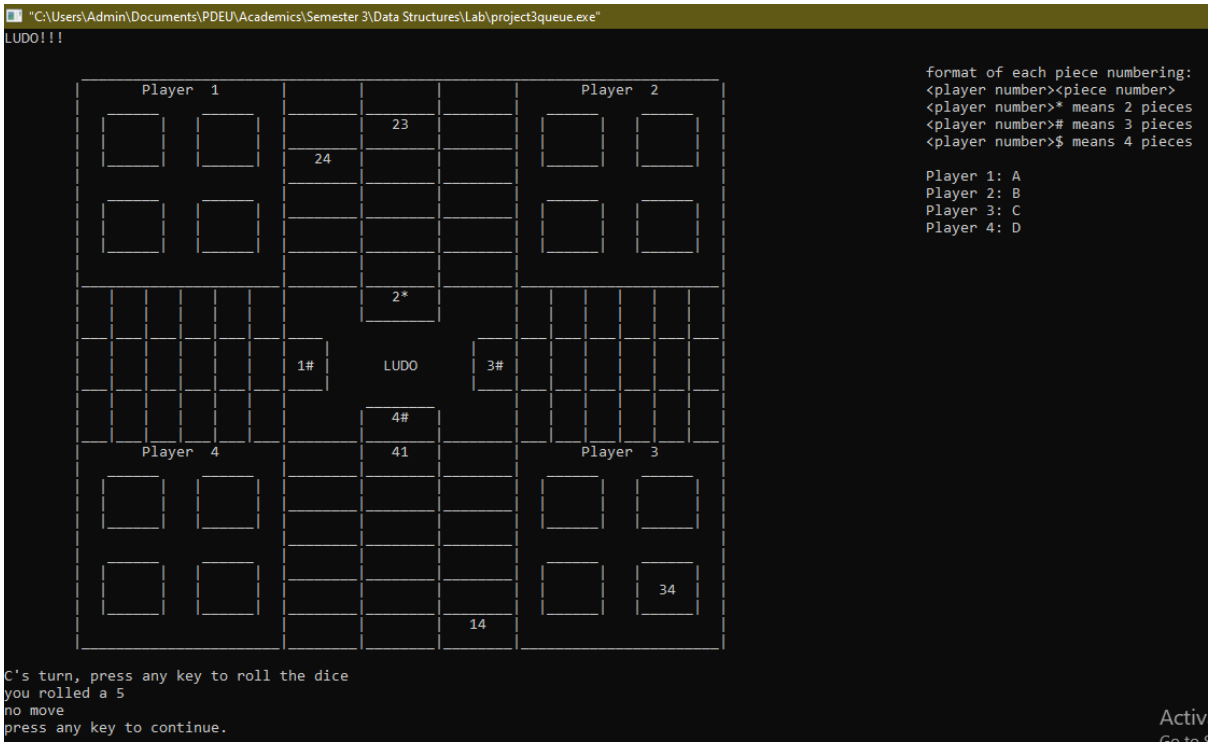

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 3
Player 4
LUDO
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
D's turn, press any key to roll the dice
you rolled a 1
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six):

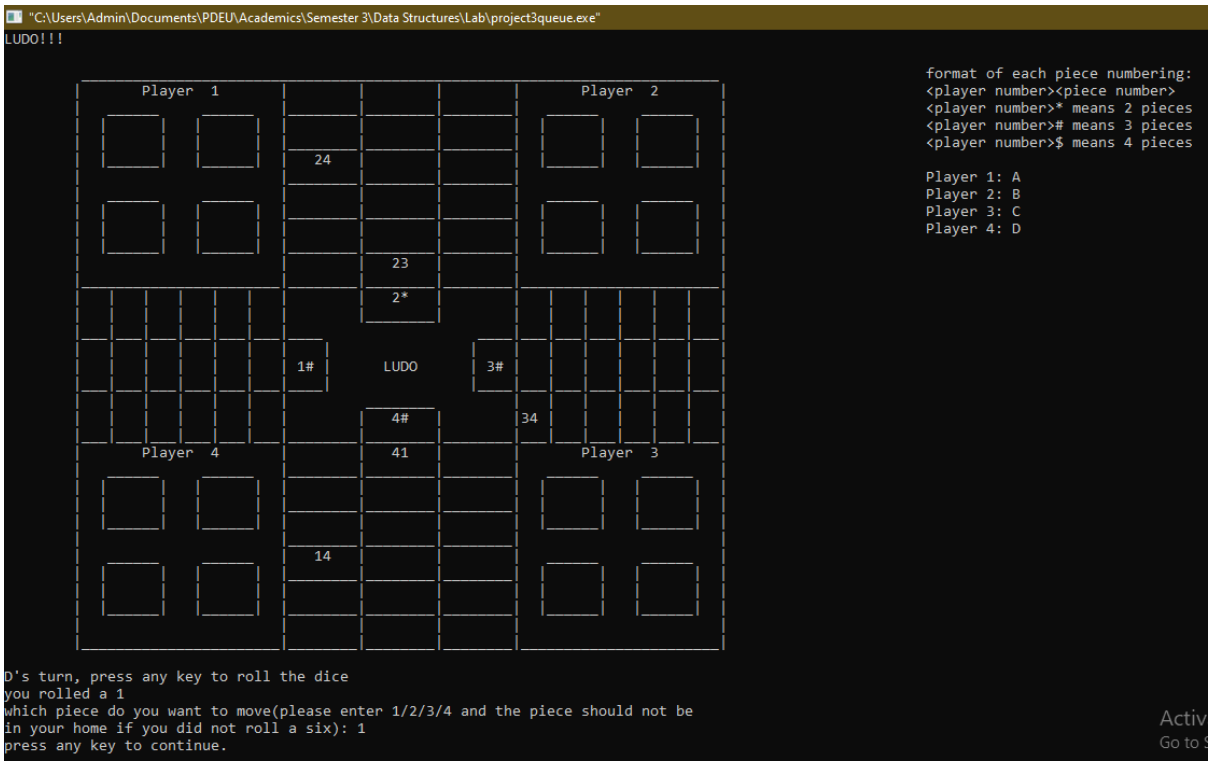

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 3
Player 4
44
14
21
31
13
24
41 43
22
11
LUDO
33
42
32
23
12
34
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
A's turn, press any key to roll the dice

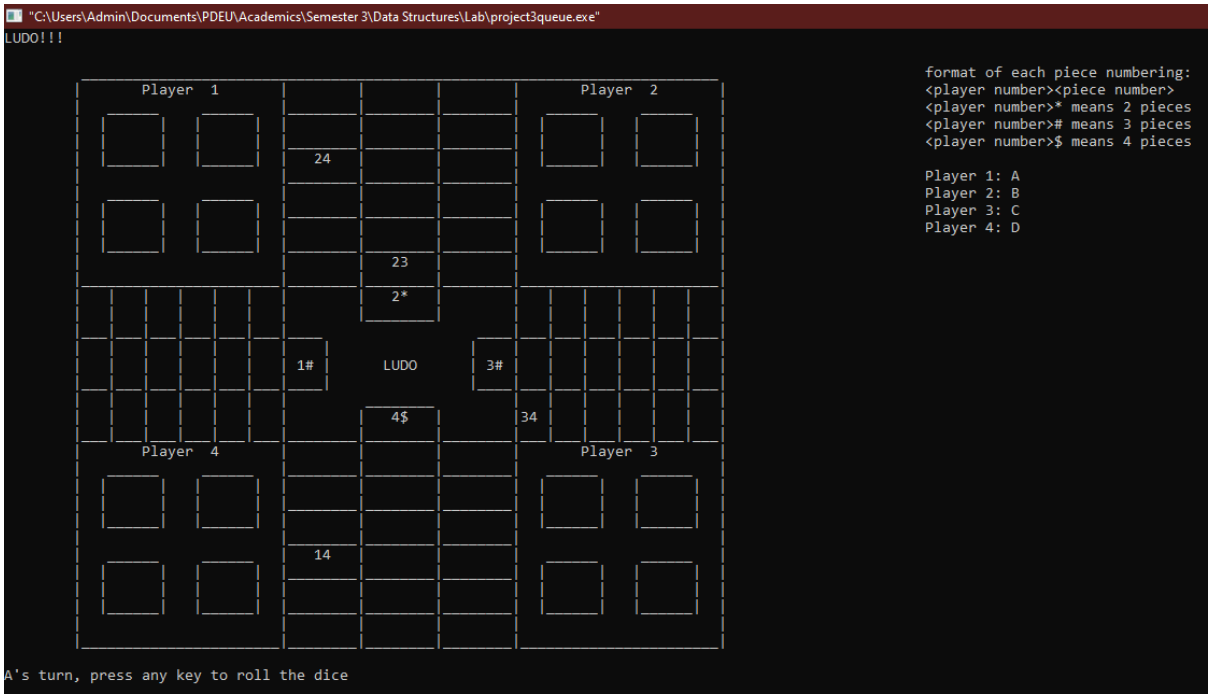

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 3
Player 4
2*
13
1*
LUDO
3*
32
4*
14
41
23
34
43
24
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
D's turn, press any key to roll the dice
you rolled a 4
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six):

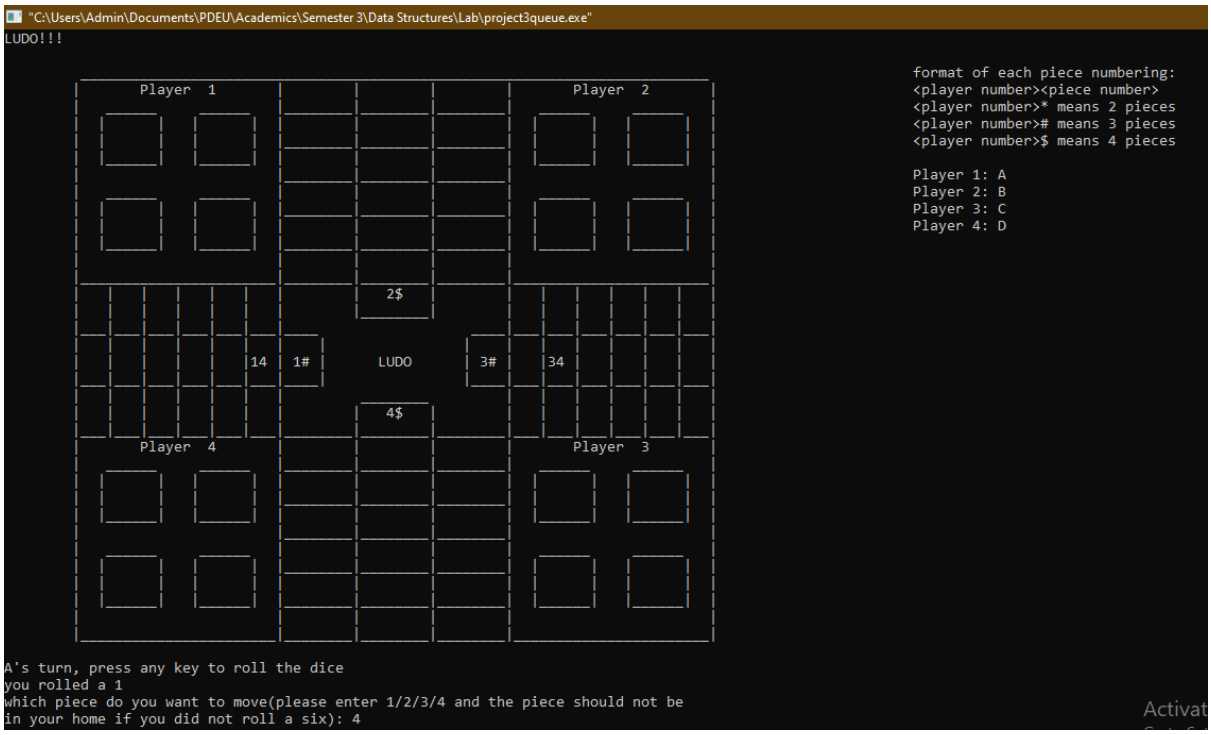

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 4
Player 3
LUDO
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
A's turn, press any key to roll the dice

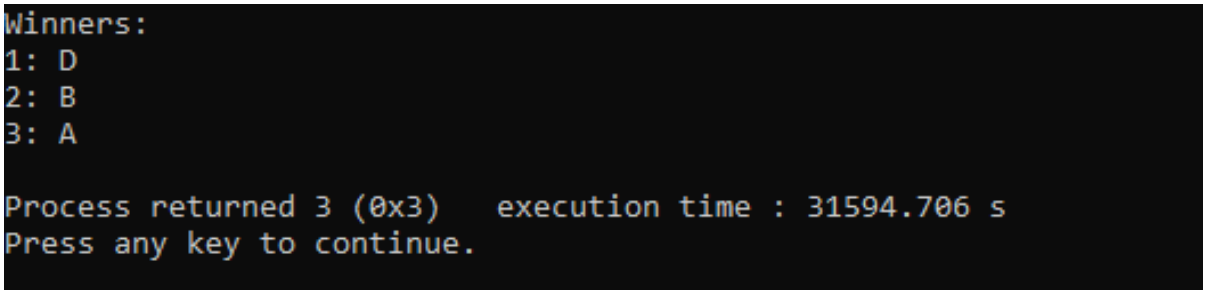

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 4
Player 3
LUDO
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
B's turn, press any key to roll the dice
you rolled a 1
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six):


"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 4
Player 3
LUDO
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
C's turn, press any key to roll the dice
you rolled a 5
no move
press any key to continue.


"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 4
Player 3
LUDO
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
D's turn, press any key to roll the dice
you rolled a 1
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six): 1
press any key to continue.


"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 4
Player 3
24
23
2*
1#
LUDO
3#
4$
34
14
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
A's turn, press any key to roll the dice


"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project3queue.exe"
LUDO!!!
Player 1
Player 2
Player 4
Player 3
2$
14
1#
LUDO
3#
34
4$
format of each piece numbering:
<player number><piece number>
<player number>* means 2 pieces
<player number># means 3 pieces
<player number>$ means 4 pieces
Player 1: A
Player 2: B
Player 3: C
Player 4: D
A's turn, press any key to roll the dice
you rolled a 1
which piece do you want to move(please enter 1/2/3/4 and the piece should not be
in your home if you did not roll a six): 4


Winners:
1: D
2: B
3: A
Process returned 3 (0x3) execution time : 31594.706 s
Press any key to continue.

Project 4: Linked List

MULTI GAMES - CHECKERS, TIC TAC TOE, SNAKES AND LADDERS, TREASURE HUNT

Code:

```
//task - create a linked list real-life applications

//MULTI GAMES - CHECKERS, TICTACTOE, SNAKES AND LADDERS, TREASURE HUNT


#include<stdio.h>

#include<string.h>

#include<math.h>

#include<stdlib.h>

#include<conio.h>


//structure to store player data

struct player

{

   int p_no;

   char name[40];

   int points;

};


//linked list-node structure

struct node

{

   struct player data;

   struct node* link;

};
```

```
//declaring some global variables
struct node* start;
char square[10] = { 'o', '1', '2', '3', '4', '5', '6', '7', '8', '9' };

//REHA

//function to create new node
struct node* new_node(char name[],int p)
{
   struct node* t;
   t=start;
   if(t!=NULL)
   {
      while(t->link!=NULL)
         t=t->link;
      t->link=(struct node*)malloc(sizeof(struct node));
      t->link->data=(struct player){.p_no=(t->data.p_no+1),.name=" ",.points=p};
      strcpy(t->link->data.name,name);
      t->link->link=NULL;
      t=t->link;
   }
   else
   {
      start=(struct node*)malloc(sizeof(struct node));
      start->data=(struct player){.p_no=1,.name=" ",.points=p};
      strcpy(start->data.name,name);
      start->link=NULL;
      t=start;
   }
   return t;
```

```
}


void ch_display(char a[8][8])
{
   printf("    y^ x-> 1   2   3   4   5   6   7   8\n");
   printf("     |  _____________________________________                If you want to end the game or the game ends:\n");
   printf("        1| %c | %c | %c | %c | %c | %c | %c | %c |                 Enter x or y or both as 0.\n",a[0][0],a[0][1],a[0][2],a[0][3],a[0][4],a[0][5],a[0][6],a[0][7]);
   printf("         |___|___|___|___|___|___|___|___|\n");
   printf("        2| %c | %c | %c | %c | %c | %c | %c | %c |\n",a[1][0],a[1][1],a[1][2],a[1][3],a[1][4],a[1][5],a[1][6],a[1][7]);
   printf("         |___|___|___|___|___|___|___|___|\n");
   printf("        3| %c | %c | %c | %c | %c | %c | %c | %c |\n",a[2][0],a[2][1],a[2][2],a[2][3],a[2][4],a[2][5],a[2][6],a[2][7]);
   printf("         |___|___|___|___|___|___|___|___|\n");
   printf("        4| %c | %c | %c | %c | %c | %c | %c | %c |\n",a[3][0],a[3][1],a[3][2],a[3][3],a[3][4],a[3][5],a[3][6],a[3][7]);
   printf("         |___|___|___|___|___|___|___|___|                 Points:\n");
   printf("        5| %c | %c | %c | %c | %c | %c | %c | %c |                 Winner: +20\n",a[4][0],a[4][1],a[4][2],a[4][3],a[4][4],a[4][5],a[4][6],a[4][7]);
   printf("         |___|___|___|___|___|___|___|___|                 Loser: -5\n");
   printf("        6| %c | %c | %c | %c | %c | %c | %c | %c |\n",a[5][0],a[5][1],a[5][2],a[5][3],a[5][4],a[5][5],a[5][6],a[5][7]);
   printf("         |___|___|___|___|___|___|___|___|\n");
   printf("        7| %c | %c | %c | %c | %c | %c | %c | %c |\n",a[6][0],a[6][1],a[6][2],a[6][3],a[6][4],a[6][5],a[6][6],a[6][7]);
   printf("         |___|___|___|___|___|___|___|___|\n");
   printf("        8| %c | %c | %c | %c | %c | %c | %c | %c |\n",a[7][0],a[7][1],a[7][2],a[7][3],a[7][4],a[7][5],a[7][6],a[7][7]);
   printf("         |___|___|___|___|___|___|___|___|\n");
}


struct node* checkp(char nm[])
```

```
{
    struct node* t;
    t=start;
    if(t==NULL)
        return t;
    while(strcmp(t->data.name,nm)!=0)
    {
        t=t->link;
        if(t==NULL)
            return t;
    }
    return t;
}

void checkers()
{
    system("cls");
    printf("WELCOME TO CHECKERS!\n");
    char name1[40],name2[40];
    char a[8][8] = {{' ',' ',' ',' ',' ',' ',' ',' '},
    {' ',' ',' ',' ',' ',' ',' ',' '},
    {' ',' ',' ',' ',' ',' ',' ',' '},
    {' ',' ',' ',' ','1','2',' ',' '},
    {' ',' ',' ',' ','2','1',' ',' '},
    {' ',' ',' ',' ',' ',' ',' ',' '},
    {' ',' ',' ',' ',' ',' ',' ',' '},
    {' ',' ',' ',' ',' ',' ',' ',' '}};
    int yt=1,turn=49,x,y,co,co1,i,j,po1=0,po2=0;
    //getting name of player 1 and creating/accessing node that stores that users info
    printf("\nEnter name of player 1 (only first 39 characters considered): ");
```

```
scanf("%39s",name1);
struct node* p1;
p1=checkp(name1);
if(p1!=NULL)
   printf("Welcome back %s, your current point score is: %d\n",name1,p1->data.points);
else
{
   p1=new_node(name1,0);
   printf("Welcome!! You are our newest member and you start with 0 points!\n");
}
//getting name of player 2 and creating/accessing node that stores that users info
printf("\nEnter name of player 2 (only first 39 characters considered): ");
scanf("%39s",name2);
struct node* p2;
p2=checkp(name2);
if(p2!=NULL)
   printf("Welcome back %s, your current point score is: %d\n",name2,p2->data.points);
else
{
   p2=new_node(name2,0);
   printf("Welcome!! You are our newest member and you start with 0 points!\n");
}
printf("\nPress any key to start the game.");
getch();
//actual game code
while(yt)
{
   system("cls");
   printf("CHECKERS GAME\n\n");
   printf("Player 1: %s\nPlayer 2: %s\n\n",name1,name2);
```

```
ch_display(a);
printf("\nturn of player %d: \n",turn-48);
//input for position to put next piece
do
{
   //taking input until the position exists and is empty
   do
   {
      printf("enter x and y for box you want:\nx(1-8): ");
      scanf("%d",&x);
      printf("\ny(1-8): ");
      scanf("%d",&y);
      if((x==0)||(y==0))
      {
         yt=0;
         break;
      }
   }while((x>8)||(x<=0)||(y>8)||(y<=0)||(a[(y-1)%9][(x-1)%9]!=' '));
   //game ends if x or y are 0
   if(yt==0)
      break;
   //checking if any conversions happen with that place and if yes, doing those
conversions
   co1=0;
   if(y!=1)
   {
      if(a[y-2][x-1]==(((turn-48)%2)+49))
      {
         co=0;
         for(i=y-3;i>=0;i--)
         {
```

```
            if(a[i][x-1]==(char)turn)
            {
               co=1;
               co1=1;
               break;
            }
         }
         if(co==1)
         {
            for(i;i<=y-2;i++)
               a[i][x-1]=(char)turn;
         }
      }
   }
   if(y!=8)
   {
      if(a[y][x-1]==(char)(((turn-48)%2)+49))
      {
         co=0;
         for(i=y+1;i<=7;i++)
         {
            if(a[i][x-1]==(char)turn)
            {
               co=1;
               co1=1;
               break;
            }
         }
         if(co==1)
         {
```

```
            for(i;i>=y;i--)
                a[i][x-1]=(char)turn;
        }
    }
}
if(x!=1)
{
    if(a[y-1][x-2]==(char)(((turn-48)%2)+49))
    {
        co=0;
        for(i=x-3;i>=0;i--)
        {
            if(a[y-1][i]==(char)turn)
            {
                co=1;
                co1=1;
                break;
            }
        }
        if(co==1)
        {
            for(i;i<=x-2;i++)
                a[y-1][i]=(char)turn;
        }
    }
}
if(x!=8)
{
    if(a[y-1][x]==(char)(((turn-48)%2)+49))
    {
```

```
            co=0;
            for(i=x+1;i<=7;i++)
            {
               if(a[y-1][i]==(char)turn)
               {
                  co=1;
                  co1=1;
                  break;
               }
            }
            if(co==1)
            {
               for(i;i>=x;i--)
                  a[y-1][i]=(char)turn;
            }
         }
      }
   }while(co1==0);
   a[y-1][x-1]=(char)turn;
   turn=((turn-48)%2)+49;
}
system("cls");
printf("CHECKERS GAME\n\n");
printf("Player 1: %s\nPlayer 2: %s\n\n",name1,name2);
ch_display(a);
for(i=0;i<8;i++)
{
   for(j=0;j<8;j++)
   {
      if(a[i][j]=='1')
```

```
            po1+=1;
        else if(a[i][j]=='2')
            po2+=1;
      }
   }
   if(po1>po2)
   {
      p1->data.points+=20;
      p2->data.points-=5;
      printf("\n%s won!!\n%s gets 20 more points and %s loses 5 points..\nSo, now,
points:\n%s: %d\n%s: %d",name1,name1,name2,name1,p1->data.points,name2,p2-
>data.points);
   }
   else if(po2>po1)
   {
      p1->data.points-=5;
      p2->data.points+=20;
      printf("\n%s won!!\n%s gets 20 more points and %s loses 5 points..\nSo, now,
points:\n%s: %d\n%s: %d",name2,name2,name1,name2,p2->data.points,name1,p1-
>data.points);
   }
   else
      printf("Its a draw!!\nSo, no change in score of either player:\n%s: %d\n%s:
%d",name1,p1->data.name,name2,p2->data.name);
   printf("\n\npress any key to go back to main page.");
   getch();
   system("cls");
}



void main()
{
```

```
    //declaring variables
    int f=1,choice;
    //accessing file where user data is stored and putting it in the linked list
    FILE* fr;
    fr=fopen("C:/Users/Admin/Documents/PDEU/Academics/Semester 3/Data Structures/Lab/User Data","r");
    start=(struct node*)malloc(sizeof(struct node));
    struct node* p;
    struct node* q;
    p=start;
    q=p;
    while(fscanf(fr,"%d %s %d\n",&p->data.p_no,p->data.name,&p->data.points)!=EOF)
    {
       p->link=(struct node*)malloc(sizeof(struct node));
       q=p;
       p=p->link;
    }
    if(p==start)
       start=NULL;
    q->link=NULL;
    free(p);
    p=NULL;
    //actual application
    while(f==1)
    {
       printf("Welcome to MANY GAMES !!\n");
       printf("\nChoose any one of these games:\n1.Treasure Hunt\n2.Checkers\n3.Snakes and Ladders\n4.Tic Tac Toe\n5.leave application\n\nNOTE: Point system and User data storage system not available in treasure hunt currently.\n\nEnter choice: ");
       scanf("%d",&choice);
       if(choice==1)
```

```
            treasure_hunt();
        else if(choice==2)
            checkers();
        else if(choice==3)
            snake();
        else if(choice==4)
            tictactoe();
        else if(choice==5)
            f=0;
        else
            printf("invalid choice\n");
    }
    printf("\nUser Data in system currently:\n\n");
    FILE* fw;
    fw=fopen("C:/Users/Admin/Documents/PDEU/Academics/Semester 3/Data
Structures/Lab/User Data","w");
    p=start;
    while(p!=NULL)
    {
        fprintf(fw,"%d %s %d\n",p->data.p_no,p->data.name,p->data.points);
        printf("%d %s %d\n",p->data.p_no,p->data.name,p->data.points);
        p=p->link;
    }
    fclose(fr);
    fclose(fw);
    printf("\nHOPE YOU HAD FUN!!\n");
}

//KRUPA AND MISARI

//function to roll dice
```

```
int rd()
{
	int rem;
	A:rem=rand()%7;
	if(rem==0)
		goto A;
	else
		return rem;
}

//function to display board
void displaychart(int curp,char player[4],struct node* p1,struct node* p2)
{	int i,j,t,c,sft=0,diceres,pos1,pos2,f=1;
		if(curp==100)
		{
			printf("Congratulations!!!!!! \n\nPlayer %s wins\n",player);
			if(player=='1')
	{
	   p1->data.points+=20;
	   p2->data.points-=5;
	   printf("\n%s gets 20 more points and %s loses 5 points..\nSo, now, points:\n%s: %d\n%s: %d",p1->data.name,p2->data.name,p1->data.name,p1->data.points,p2->data.name,p2->data.points);
	}
	else
	{
	   p2->data.points+=20;
	   p1->data.points-=5;
	   printf("\n%s gets 20 more points and %s loses 5 points..\nSo, now, points:\n%s: %d\n%s: %d",p2->data.name,p1->data.name,p2->data.name,p2->data.points,p1->data.name,p1->data.points);
	}
```

```
        scanf("%*s");
        f=0;
      }
   if(f==1)
   {
        for(i=10;i>0;i--)
        {
                t=i-1;
                if((sft%2)==0)
                {
                        c=0;
                        for(j=10;j>=1;j--)
                        {
                                diceres=(i*j)+(t*c++);
                                if(curp==diceres)
                                        printf("%s\t",player);
                                else
                                printf("%d\t",diceres);
                        }
                        sft++;
                }
                else
                {
                        c=9;
                        for(j=1;j<=10;j++)
                        {
                                diceres=(i*j)+(t*c--);
                                if(curp==diceres)
                                        printf("%s\t",player);
                                else
```

```
                                        printf("%d\t",diceres);
                        }
                        sft++;
                }
                printf("\n\n");
        }

printf("\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd
\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xc
d\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\x
cd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n");
    }
}



//snakes and ladders game
void snake()
{
    system("cls");
    char mark,name1[40],name2[40];
    //getting name of player 1 and creating/accessing node that stores that users info
    printf("\nEnter name of player 1 (only first 39 characters considered): ");
    scanf("%39s",name1);
    struct node* p1;
    p1=checkp(name1);
    if(p1!=NULL)
        printf("Welcome back %s, your current point score is: %d\n",name1,p1->data.points);
    else
    {
        p1=new_node(name1,0);
        printf("Welcome!! You are our newest member and you start with 0 points!\n");
    }
```

```
    //getting name of player 2 and creating/accessing node that stores that users info
    printf("\nEnter name of player 2 (only first 39 characters considered): ");
    scanf("%39s",name2);
    struct node* p2;
    p2=checkp(name2);
    if(p2!=NULL)
       printf("Welcome back %s, your current point score is: %d\n",name2,p2->data.points);
    else
    {
       p2=new_node(name2,0);
       printf("Welcome!! You are our newest member and you start with 0 points!\n");
    }
    printf("\nPress any key to start the game.");
    getch();
    system("cls");
          int i,dice,cpos1=0,cpos2=0,l=1;
          char ch;
          while(l)
          {
                printf("Snakes: | 25 to 9  | 65 to 40 | 99 to 1  |\nLadder: | 13 to 42 | 60 to 83 | 70
to 93 |\n\n");
                printf("Choose your option\n");
                printf("[1] Player 1 plays\n");
                printf("[2] Player 2 plays\n");
                printf("[3] Exit\n");
                scanf("%s",&ch);

                switch(ch)
                {

                      case '1':dice=rd();
```

```
                system("cls");

printf("\n\n\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\
xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd
\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xc
d\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n");
                printf("\t\t\t    Snakes And Ladders\n");

printf("\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd
\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xc
d\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\x
cd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n");
                        cpos1=dice+cpos1;
                        if(cpos1<101){
                                if(cpos1==99)
        displaychart(1,"-P1-",p1,p2);//snake
                                if(cpos1==65)
        displaychart(40,"-P1-",p1,p2);//snake
                                if(cpos1==25)
        displaychart(9,"-P1-",p1,p2);//snake
                                if(cpos1==70)
        displaychart(93,"-P1-",p1,p2);//ladder
                                if(cpos1==60)
        displaychart(83,"-P1-",p1,p2);//ladder
                                if(cpos1==13)
        displaychart(42,"-P1-",p1,p2);//ladder
                                else
                                        displaychart(cpos1,"-P1-",p1,p2);
            printf("\t\t\t\tDice = %d\n",dice);

printf("\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd
\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xc
d\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\x
cd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n");
                        }
```

```
                                else{
                                        cpos1=cpos1-dice;
                                        printf("Range exceeded of Player 1.\n");
                                        displaychart(cpos1,"-P1-",p1,p2);
                                }
                                printf("P2 is at %d\n\n",cpos2);


                        break;
                case '2':dice=rd();
        system("cls");

printf("\n\n\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\
xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd
\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xc
d\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n");
        printf("\t\t\t    Snakes And Ladders\n");

printf("\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd
\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xc
d\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\x
cd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n");
                                cpos2=dice+cpos2;
                                if(cpos2<101){
                                        if(cpos2==99) //snake
            displaychart(1,"-P2-",p1,p2);
                                        if(cpos2==65) //snake
            displaychart(40,"-P2-",p1,p2);
                                        if(cpos2==25) //snake
            displaychart(9,"-P2-",p1,p2);
                                        if(cpos2==70) //ladder
            displaychart(93,"-P2-",p1,p2);
                                        if(cpos2==60) //ladder
            displaychart(83,"-P2-",p1,p2);
```

```
                                        if(cpos2==13) //ladder
                displaychart(42,"-P2-",p1,p2);
                                        else
                                                displaychart(cpos2,"-P2-",p1,p2);
                                        printf("\t\t\t\tDice = %d\n",dice);

        printf("\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\
xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd
\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xc
d\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\xcd\n");
                                    }

                                    else{
                                        cpos2=cpos2-dice;
                                        printf("Range exceeded of Player 2.\n");
                                        displaychart(cpos2,"-P2-",p1,p2);
                                    }
                                    printf("P 1 is at %d\n\n",cpos1);
                            break;
                    case '3':l=0;
                            break;

                    default:printf("Choice is incorrect, Please Try Again:)\n");

            }

    }
}

//TANYA

//main code for treasure hunt
```

```
void treasure_hunt()
{
   system("cls");
   printf("Welcome to the hunt!\n");
   printf("Just follow the hints and collect your exciting treats.\n");
   struct nodes
   {
      char data[200];
      struct nodes* next;
   };
   /* Initialize nodes */
   struct nodes *head;
   struct nodes *one = NULL;
   struct nodes *two = NULL;
   struct nodes *three = NULL;
   struct nodes *four=NULL;
   struct nodes *five=NULL;
   struct nodes *six= NULL;
   /* Allocate memory */
   one =  malloc(sizeof(struct nodes));
   two = malloc(sizeof(struct nodes));
   three = malloc(sizeof(struct nodes));
   four= malloc(sizeof(struct nodes));
   five=malloc(sizeof(struct nodes));
   six=malloc(sizeof(struct nodes));
   /* Assign data values */
   strcpy(one->data , "if you are in a hungry mood,go herefirst and find some food");
   strcpy(two->data ," Now you are on your second clue,these go on before your shoes");
   strcpy(three->data, "If you want your teeth to shine,pick this up and spend some time");
   strcpy(four->data, "If you want to learn and grow, turn the page,get in the know");
```

```
    strcpy(five->data,"Add some colour to your days!Pick these up,you are on the way");
    strcpy(six->data, "Take a walk and step outside,this is where you go to ride");
    //displaying clues
        printf("First hint is: %s\n", (one->data));
        printf("Second hint is: %s\n", (two->data));
        printf("Third hint is: %s\n", (three->data));
        printf("Fourth hint is: %s\n", (four->data));
        printf("Fifth hint is: %s\n", (five->data));
        printf("Sixth hint is: %s\n", (six->data));
        printf("\n\npress any key to exit.");
        getch();
        system("cls");
}

//VRUNDA

//main part of code of tictactoe
void tictactoe()
{
    system("cls");
    //declaring most variables
    int player = 1, i, choice;
    char mark,name1[40],name2[40];;
    //getting name of player 1 and creating/accessing node that stores that users info
    printf("\nEnter name of player 1 (only first 39 characters considered): ");
    scanf("%39s",name1);
    struct node* p1;
    p1=checkp(name1);
    if(p1!=NULL)
       printf("Welcome back %s, your current point score is: %d\n",name1,p1->data.points);
```

```
    else
    {
        p1=new_node(name1,0);
        printf("Welcome!! You are our newest member and you start with 0 points!\n");
    }
    //getting name of player 2 and creating/accessing node that stores that users info
    printf("\nEnter name of player 2 (only first 39 characters considered): ");
    scanf("%39s",name2);
    struct node* p2;
    p2=checkp(name2);
    if(p2!=NULL)
        printf("Welcome back %s, your current point score is: %d\n",name2,p2->data.points);
    else
    {
        p2=new_node(name2,0);
        printf("Welcome!! You are our newest member and you start with 0 points!\n");
    }
    printf("\nPress any key to start the game.");
    getch();
    //actual game code
    do
    {
        board();
        printf("Information:\nPlayer 1 (X): %s\nPlayer 2 (O): %s\n\nPoint
distribution:\nWinner: +20\nLoser: -5\nDraw: +0 for both\n\n",name1,name2);
        player = (player % 2) ? 1 : 2;
        printf("Player %d, enter a number:  ", player);
        scanf("%d", &choice);

        mark = (player == 1) ? 'X' : 'O';
```

```
if (choice == 1 && square[1] == '1')
    square[1] = mark;

else if (choice == 2 && square[2] == '2')
    square[2] = mark;

else if (choice == 3 && square[3] == '3')
    square[3] = mark;

else if (choice == 4 && square[4] == '4')
    square[4] = mark;

else if (choice == 5 && square[5] == '5')
    square[5] = mark;

else if (choice == 6 && square[6] == '6')
    square[6] = mark;

else if (choice == 7 && square[7] == '7')
    square[7] = mark;

else if (choice == 8 && square[8] == '8')
    square[8] = mark;

else if (choice == 9 && square[9] == '9')
    square[9] = mark;

else
{
    printf("Invalid move. Please press ant key!!");
```

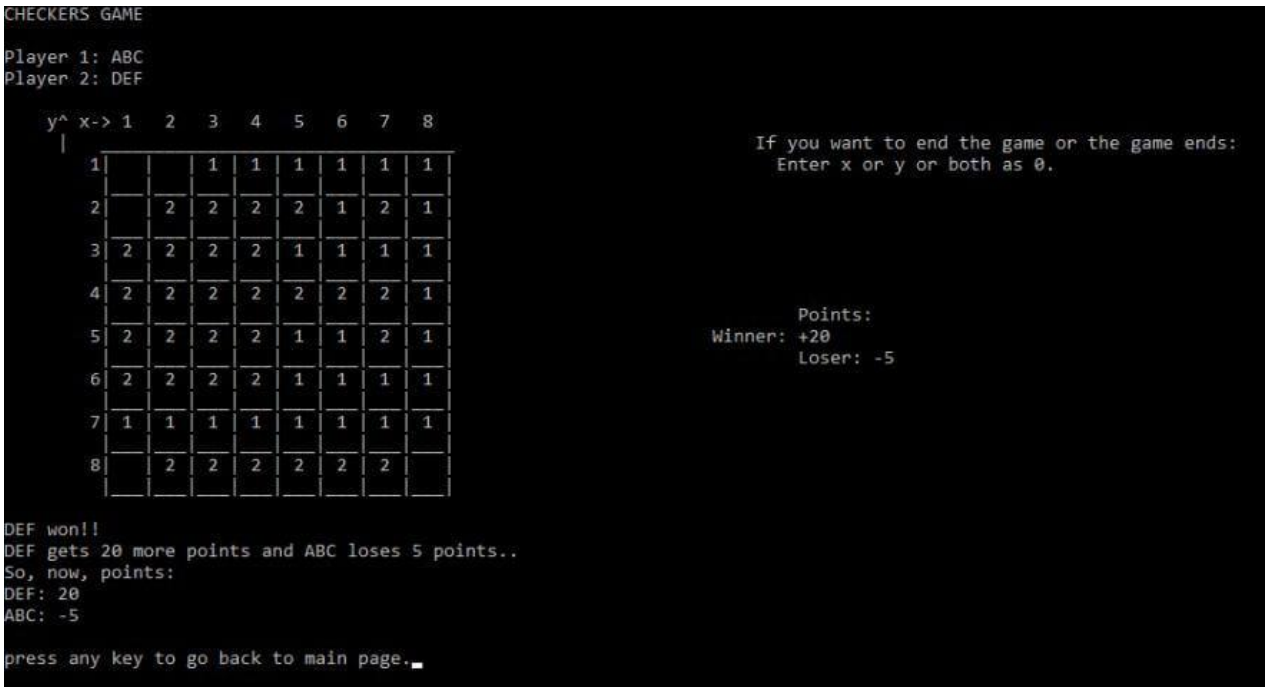
```
        player--;player = (player % 2) ? 1 : 2;
        getch();
      }
      i = checkwin();
      player++;
   }while (i ==  - 1);
   board();
   if (i == 1)
   {
      printf("==>\aPlayer %d win\n", --player);
      if(player==1)
      {
         p1->data.points+=20;
         p2->data.points-=5;
         printf("\n%s gets 20 more points and %s loses 5 points..\nSo, now, points:\n%s:
%d\n%s: %d",name1,name2,name1,p1->data.points,name2,p2->data.points);
      }
      else
      {
         p2->data.points+=20;
         p1->data.points-=5;
         printf("\n%s gets 20 more points and %s loses 5 points..\nSo, now, points:\n%s:
%d\n%s: %d",name2,name1,name2,p2->data.points,name1,p1->data.points);
      }
   }
   else
   {
      printf("==>\aGame draw\n");
      printf("\nSo, no change in score of either player:\n%s: %d\n%s: %d",name1,p1-
>data.name,name2,p2->data.name);
```

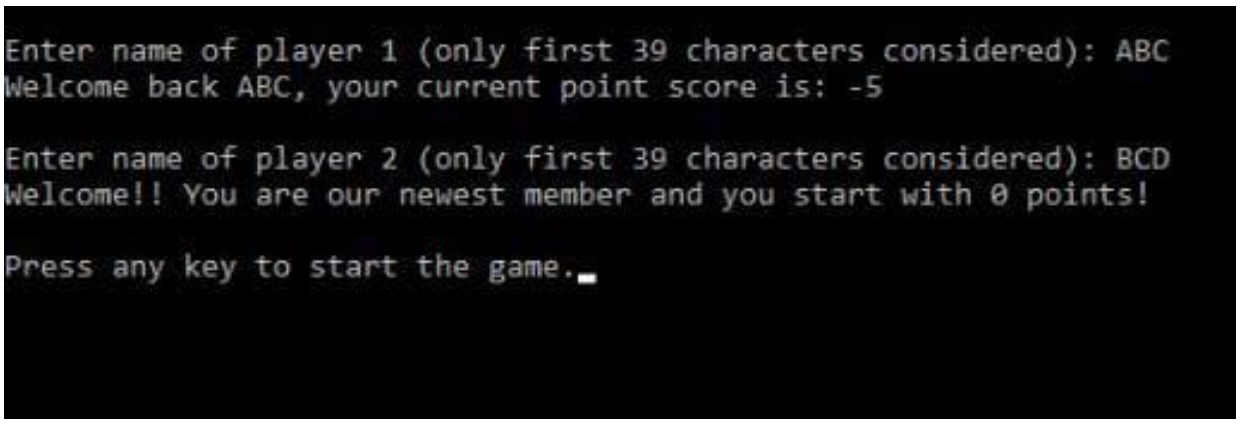
```
    }
    printf("\n\nPress any key to go back to home page.");
    getch();
    system("cls");
}

//function to check who won
int checkwin()
{
    if (square[1] == square[2] && square[2] == square[3])
        return 1;   //1 FOR GAME IS OVER WITH RESULT

    else if (square[4] == square[5] && square[5] == square[6])
        return 1;

    else if (square[7] == square[8] && square[8] == square[9])
        return 1;

    else if (square[1] == square[4] && square[4] == square[7])
        return 1;

    else if (square[2] == square[5] && square[5] == square[8])
        return 1;

    else if (square[3] == square[6] && square[6] == square[9])
        return 1;

    else if (square[1] == square[5] && square[5] == square[9])
        return 1;
```

```
    else if (square[3] == square[5] && square[5] == square[7])
        return 1;

    else if (square[1] != '1' && square[2] != '2' && square[3] != '3' &&
        square[4] != '4' && square[5] != '5' && square[6] != '6' && square[7]
        != '7' && square[8] != '8' && square[9] != '9')

        return 0;                                   //O GAME IS OVER AND NO RESULT
    else
        return  - 1;                                //-1 FOR GAME IS IN PROGRESS
}

//function to print board
void board()
{
    system("cls");
    printf("\n\n\tTic Tac Toe\n\n");
    printf("Player 1 (X)  -  Player 2 (O)\n\n\n");
    printf("     |     |     \n");
    printf("  %c  |  %c  |  %c \n", square[1], square[2], square[3]);
    printf("_____|_____|_____\n");
    printf("     |     |     \n");
    printf("  %c  |  %c  |  %c \n", square[4], square[5], square[6]);
    printf("_____|_____|_____\n");
    printf("     |     |     \n");
    printf("  %c  |  %c  |  %c \n", square[7], square[8], square[9]);
    printf("     |     |     \n\n");
}
```

Output:


Welcome to MANY GAMES !!
Choose any one of these games:
1.Treasure Hunt
2.Checkers
3.Snakes and Ladders
4.Tic Tac Toe
5.leave application
NOTE: Point system and User data storage system not available in treasure hunt currently.
Enter choice:

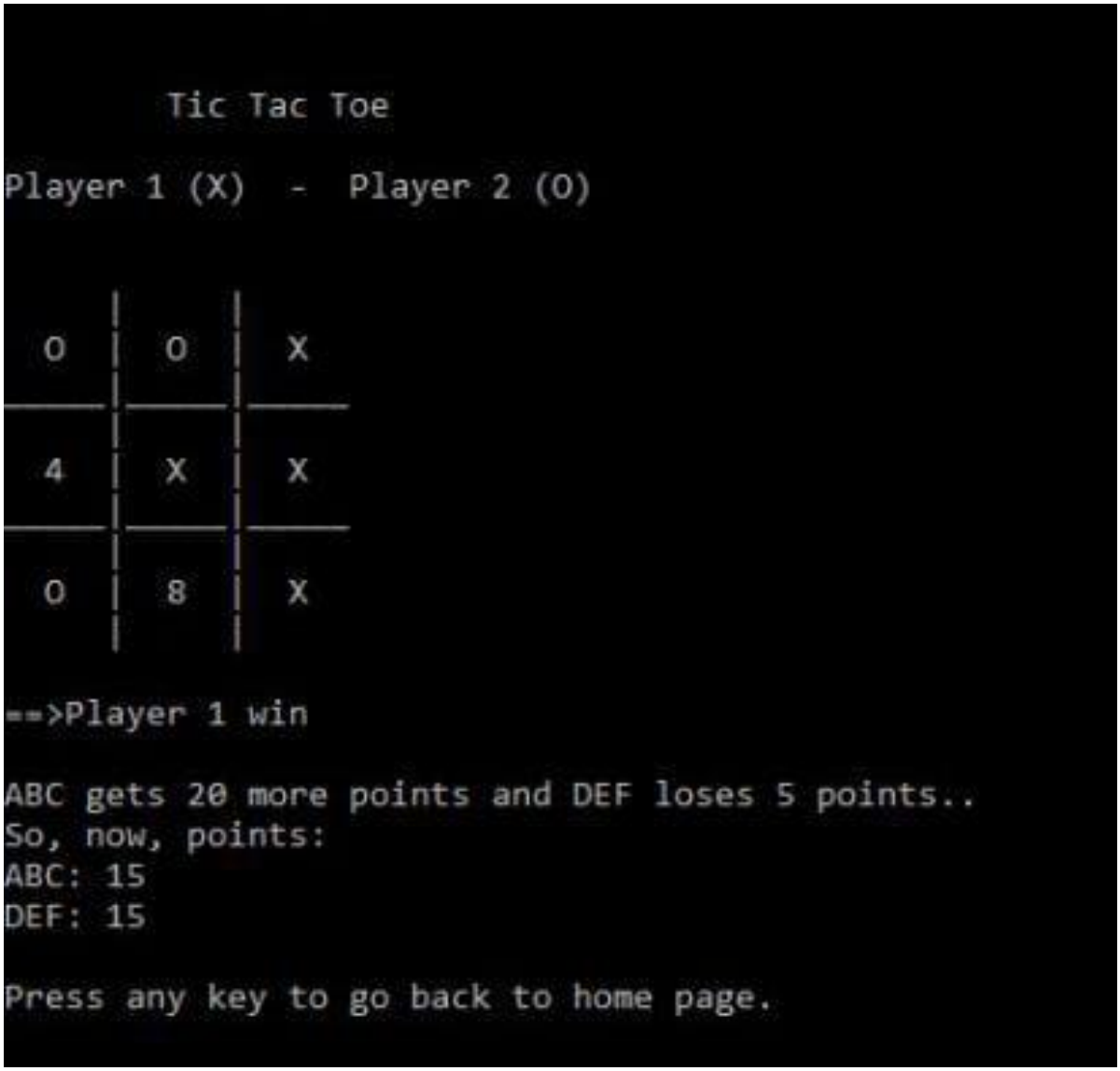

CHECKERS GAME
Player 1: ABC
Player 2: DEF
y^ x-> 1 2 3 4 5 6 7 8
If you want to end the game or the game ends:
Enter x or y or both as 0.
Points:
Winner: +20
Loser: -5
DEF won!!
DEF gets 20 more points and ABC loses 5 points..
So, now, points:
DEF: 20
ABC: -5
press any key to go back to main page.

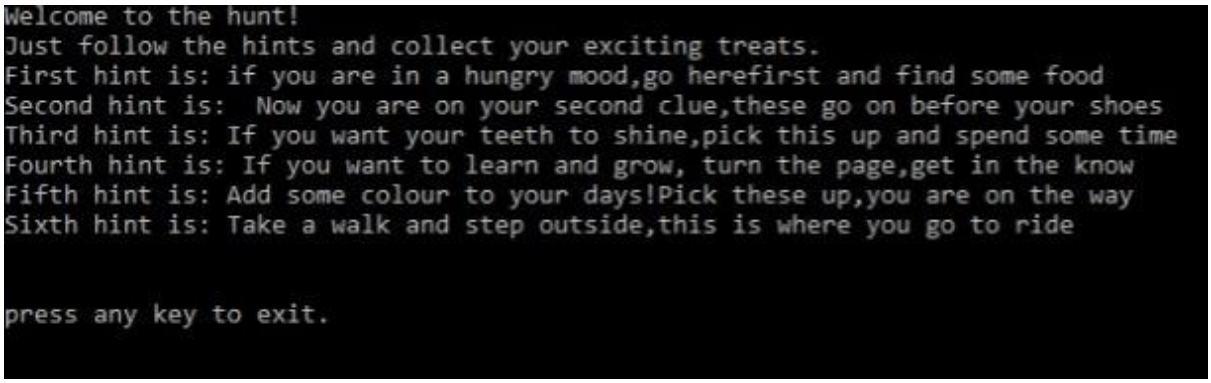

Enter name of player 1 (only first 39 characters considered): ABC
Welcome back ABC, your current point score is: -5
Enter name of player 2 (only first 39 characters considered): BCD
Welcome!! You are our newest member and you start with 0 points!
Press any key to start the game.

```
Snakes: | 25 to 9  | 65 to 40 | 99 to 1  |
Ladder: | 13 to 42 | 60 to 83 | 70 to 93 |

Choose your option
[1] Player 1 plays
[2] Player 2 plays
[3] Exit
```

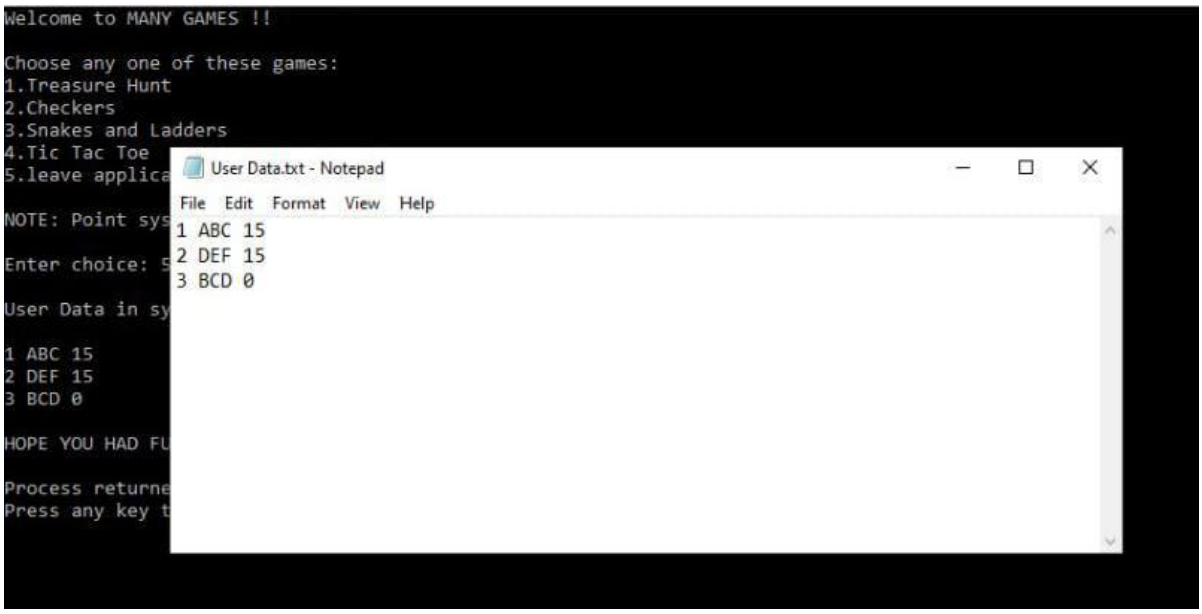
```
80      79      78      77      76      75      74      73      72      71
61      62      63      64      65      66      67      68      69      70
60      59      58      57      56      55      54      53      52      51
41      42      43      44      45      46      47      48      49      50
40      39      38      37      36      35      34      33      32      31
21      22      23      24      25      26      27      28      29      30
20      19      18      17      16      15      14      13      -P1-    11
1       2       3       4       5       6       7       8       9       10
_____________________________________________________________________________
                                Dice = 6
_____________________________________________________________________________
P2 is at 1

Snakes: | 25 to 9  | 65 to 40 | 99 to 1  |
Ladder: | 13 to 42 | 60 to 83 | 70 to 93 |

Choose your option
[1] Player 1 plays
[2] Player 2 plays
[3] Exit
```

```
Enter name of player 1 (only first 39 characters considered): ABC
Welcome back ABC, your current point score is: -5

Enter name of player 2 (only first 39 characters considered): DEF
Welcome back DEF, your current point score is: 20

Press any key to start the game.
```


Tic Tac Toe
Player 1 (X) - Player 2 (O)
O | O | X
4 | X | X
O | 8 | X
==>Player 1 win
ABC gets 20 more points and DEF loses 5 points..
So, now, points:
ABC: 15
DEF: 15
Press any key to go back to home page.


Welcome to the hunt!
Just follow the hints and collect your exciting treats.
First hint is: if you are in a hungry mood,go herefirst and find some food
Second hint is: Now you are on your second clue,these go on before your shoes
Third hint is: If you want your teeth to shine,pick this up and spend some time
Fourth hint is: If you want to learn and grow, turn the page,get in the know
Fifth hint is: Add some colour to your days!Pick these up,you are on the way
Sixth hint is: Take a walk and step outside,this is where you go to ride
press any key to exit.

```
Welcome to MANY GAMES !!

Choose any one of these games:
1.Treasure Hunt
2.Checkers
3.Snakes and Ladders
4.Tic Tac Toe
5.leave application

NOTE: Point system and User data storage system not available in treasure hunt currently.

Enter choice: 5

User Data in system currently:

1 ABC 15
2 DEF 15
3 BCD 0

HOPE YOU HAD FUN!!

Process returned 0 (0x0)   execution time : 835.922 s
Press any key to continue.
```



Project 5: Structure, Stack, Queue, Linked List

HOSPITAL PATIENT INFORMATION DATABASE

Code:

```
//task - create a linked list, queue, stack and structure real-life applications
//HOSPITAL PATIENT INFORMATION DATABASE
//queues-patients for today-new ones dequeued into main database at the end of the day and for old ones data is updated-maximum limit 100
//different queue for different departments of the hospital-according to the patients current illness
//stack-patient health portfolio-old first, new last
//linked list-stores data from database

#include<stdio.h>
#include<stdlib.h>
#include<string.h>

//structure to store data in stack
struct data_s
{
   char symptoms[400];
   char diagnosis[100];
   char type[50];
   char treatment[500];
};

//structure for stack
struct node_s
{
   struct data_s data;
```

```
    struct node_s* link;
};

//structure to store patient info
struct patient
{
    struct node_s* start;
    int p_no;
    char name[60];
    char sex[10];
    int age;
    long int phone;
};

//structure for linked list node
struct node
{
    struct patient data;
    struct node* link;
};

//declaring some global variables
struct node* start;
int qp[10],qn[10],qc[10],qg[10],qy[10],fp=-1,rp=-1,fn=-1,rn=-1,fc=-1,rc=-1,fg=-1,rg=-1,fy=-
1,ry=-1;

//declaring functons
struct node_s* pop(struct node*);
struct node* new_node(struct node_s*,int,char[],char[],int,long int);
struct node* traverse(int);
void new_patient();
```

```
void p_data();
void p_change();
void r_upd();
void push(struct node*,char[],char[],char[],char[]);
void add_record();
void see_queue();
void enqueue(int,int[],int*,int*);
void dequeue(int[],int*,int*);
void display(int[],int,int);
void add_queue();
void out_queue();
void see_stack();

//REHA

//main function
void main()
{
   int f=1,c;
   //accessing file where user data is stored and putting it in the linked list
   FILE* fr;
   fr=fopen("C:/Users/Admin/Documents/PDEU/Academics/Semester 3/Data
Structures/Lab/Patient Data","r");
   start=(struct node*)malloc(sizeof(struct node));
   struct node* p;
   struct node* q;
   p=start;
   q=p;
   struct node_s* t;
   struct node_s* v;
```

```
   while(fscanf(fr,"%d %d %s %s %d %ld\n",&p->data.start,&p->data.p_no,p->data.name,p->data.sex,&p->data.age,&p->data.phone)!=EOF)
   {
      p->data.start=(struct node_s*)malloc(sizeof(struct node_s));
      t=p->data.start;
      do
      {
         t->link=(struct node_s*)malloc(sizeof(struct node_s));
         fscanf(fr,"%s %s %s %s\n",t->data.symptoms,t->data.diagnosis,t->data.type,t->data.treatment);
         v=t;
         t=t->link;
      }while(!((strcmp(v->data.symptoms,"0"))||(strcmp(v->data.diagnosis,"0"))||(strcmp(v->data.type,"0"))||(strcmp(v->data.treatment,"0"))));
      t=NULL;
      p->link=(struct node*)malloc(sizeof(struct node));
      q=p;
      p=p->link;
   }
   if(p==start)
      start=NULL;
   q->link=NULL;
   free(p);
   p=NULL;
   while(f)
   {
      printf("\nDAY START AT XYZ HOSPITAL\n\n");
      printf("Choose one of the following options to access the patient database:\n1.Create new patient record\n2.Add new entry in patient's file\n3.Add patient to queue\n4.See patient's file record\n5.See available patient information\n6.Update patient information details\n7.Update latest entry in patient file\n8.Dequeue patient\n9.See current queue for specific department\n10.Leave application\nenter choice: ");
      scanf("%d",&c);
```

```
        if(c==1)
            new_patient();
        else if(c==2)
            add_record();
        else if(c==3)
            add_queue();
        else if(c==4)
            see_stack();
        else if(c==5)
            p_data();
        else if(c==6)
            p_change();
        else if(c==7)
            r_upd();
        else if(c==8)
            out_queue();
        else if(c==9)
            see_queue();
        else if(c==10)
            f=0;
        else
            printf("invalid choice");
    }
    printf("\nUser Data in system currently:\n\n");
    FILE* fw;
    fw=fopen("C:/Users/Admin/Documents/PDEU/Academics/Semester 3/Data
Structures/Lab/Patient Data","w");
    p=start;
    while(p!=NULL)
    {
```

```
        fprintf(fw,"%d %d %s %s %d %ld\n",p->data.start,p->data.p_no,p->data.name,p->data.sex,p->data.age,p->data.phone);
        printf("%d %d %s %s %d %ld\n",p->data.start,p->data.p_no,p->data.name,p->data.sex,p->data.age,p->data.phone);
        t=p->data.start;
        while(t!=NULL)
        {
            fprintf(fw,"%s %s %s %s\n",t->data.symptoms,t->data.diagnosis,t->data.type,t->data.treatment);
            printf("%s %s %s %s\n",t->data.symptoms,t->data.diagnosis,t->data.type,t->data.treatment);
            t=t->link;
        }
        char stz[2]="0";
        fprintf(fw,"%s %s %s %s\n",stz,stz,stz,stz);
        printf("%s %s %s %s\n",stz,stz,stz,stz);
        p=p->link;
    }
    fclose(fr);
    fclose(fw);
    printf("DAY ENDS AT XYZ HOSPITAL");
}

//function to create new node
struct node* new_node(struct node_s* starts,int p_no,char name[60],char sex[10],int age,long int phone)
{
    struct node* t;
    t=start;
    if(t!=NULL)
    {
        while(t->link!=NULL)
```

```
            t=t->link;
        t->link=(struct node*)malloc(sizeof(struct node));
        t->link->data=(struct patient){.start=starts,.p_no=p_no,.name=" ",.sex="
",.age=age,.phone=phone};
        strcpy(t->link->data.name,name);
        strcpy(t->link->data.sex,sex);
        t->link->link=NULL;
        t=t->link;
    }
    else
    {
        start=(struct node*)malloc(sizeof(struct node));
        start->data=(struct patient){.start=starts,.p_no=p_no,.name=" ",.sex="
",.age=age,.phone=phone};
        strcpy(start->data.name,name);
        strcpy(start->data.sex,sex);
        start->link=NULL;
        t=start;
    }
    return t;
}

//function to find particular patient record using the patient number
struct node* traverse(int number)
{
    struct node* t;
    t=start;
    while(t!=NULL)
    {
        if(t->data.p_no==number)
            return t;
```

```
        t=t->link;
    }
    return NULL;
}

//function to add new patient in linked list
void new_patient()
{
    char n[60],s[10];
    int r,pn,a;
    printf("\nenter the following details to create a new account for the patient: \n");
    printf("name: ");
    scanf("%s",n);
    printf("age: ");
    scanf("%d",&a);
    printf("sex: ");
    scanf("%s",s);
    printf("phone number: ");
    scanf("%ld",&pn);
    struct node* t;
    t=start;
    if(t==NULL)
        r=1;
    else
    {
        while(t->link!=NULL)
        {
            t=t->link;
        }
        r=t->data.p_no+1;
```

```
    }
    struct node_s* stack;
    stack=NULL;
    t=new_node(stack,r,n,s,a,pn);
    printf("\nnew patient entry has been created and the patient number for %s with id
%d.\n\n",t->data.name,t->data.p_no);
}

//function to see patient info stored in linked list
void p_data()
{
    int n;
    printf("enter patient number: ");
    scanf("%d",&n);
    struct node* p;
    p=traverse(n);
    if(p!=NULL)
    {
        printf("\nPatient data:\nName: %s\nAge: %d\nSex: %s\nPhone Number: %ld\n",p-
>data.name,p->data.age,p->data.sex,p->data.phone);
    }
    else
        printf("patient number is incorrect.\n");
}

//function to update patient information
void p_change()
{
    int n,ch;
    printf("enter patient number: ");
    scanf("%d",&n);
```

```
    struct node* p;
    p=traverse(n);
    if(p!=NULL)
    {
        printf("\nWhich value do you want to update:\n1.Name\n2.Age\n3.Sex\n4.Phone
Number\n\nenter choice: ");
        scanf("%d",&ch);
        if(ch==1)
        {
            printf("\nenter updated name: ");
            scanf("%s",p->data.name);
        }
        else if(ch==2)
        {
            printf("\nenter updated age: ");
            scanf("%d",&p->data.age);
        }
        else if(ch==3)
        {
            printf("\nenter updated sex: ");
            scanf("%s",p->data.sex);
        }
        else if(ch==4)
        {
            printf("\nenter updated phone number: ");
            scanf("%ld",&p->data.phone);
        }
        else
            printf("\ninvalid choice.\n");
    }
    else
```

```
        printf("patient number is incorrect.\n");
}

//function to update latest stack record in patient file
void r_upd()
{
   int n,ch,d;
   printf("enter patient number: ");
   scanf("%d",&n);
   struct node* p;
   p=traverse(n);
   if(p!=NULL)
   {
      char s[400],g[100],t[50],r[500];
      struct node_s* k;
      k=pop(p);
      if(k!=NULL)
      {
      printf("\nWhich value do you want to update:\n1.Symptoms\n2.Diagnosis\n3.Hospital Department\n4.Treatment Suggested\n\nenter choice: ");
      scanf("%d",&ch);
      if(ch==1)
      {
         printf("\nenter updated symptoms: ");
         scanf("%s",k->data.symptoms);
      }
      else if(ch==2)
      {
         printf("\nenter updated diagnosis: ");
         scanf("%s",k->data.diagnosis);
      }
```

```
    else if(ch==3)
    {
        printf("\nchoose updated department:
\n1.Pediatric\n2.Neurology\n3.Cardiology\n4.General Physician\n5.Gynecology\nenter
choice: ");
        scanf("%d",&d);
        if(d==1)
            strcpy(k->data.type,"Pediatrician");
        else if(d==2)
            strcpy(k->data.type,"Neurology");
        else if(d==3)
            strcpy(k->data.type,"Cardiology");
        else if(d==4)
            strcpy(k->data.type,"General Physician");
        else if(d==5)
            strcpy(k->data.type,"Gynecology");
        else
            printf("invalid choice\n");
    }
    else if(ch==4)
    {
        printf("\nenter updated treatment: ");
        scanf("%s",k->data.treatment);
    }
    else
        printf("\ninvalid choice.\n");
    push(p,k->data.symptoms,k->data.diagnosis,k->data.type,k->data.treatment);
  }
  else
    printf("file is empty.\n");
  }
```

```
    else
        printf("patient number is incorrect.\n");
}

//TANYA

//function to push onto the stack
void push(struct node* p,char symptoms[400],char diagnosis[100],char type[50],char
treatment[500])
{
        struct node_s* n=(struct node_s*)malloc(sizeof(struct node_s));
    strcpy(n->data.symptoms,symptoms);
    strcpy(n->data.diagnosis,diagnosis);
    strcpy(n->data.type,type);
    strcpy(n->data.treatment,treatment);
        n->link=p->data.start;
        p->data.start=n;
}

//function to add an entry to the patient file stack
void add_record()
{
    int num,c;
        printf("Enter patient no.: ");
        scanf("%d",&num);
        struct node* p;
        p=traverse(num);
        if(p!=NULL)
    {
                char s[400],d[100],t[50]="none",tr[500];
                printf("Enter these details to add new record:\n");
```

```
                printf("symptoms: ");
                scanf("%s",s);
                printf("diagnosis: ");
                scanf("%s",d);
                printf("choose department:
\n1.Pediatric\n2.Neurology\n3.Cardiology\n4.General Physician\n5.Gynecology\nenter
choice: ");
        scanf("%d",&c);
        if(c==1)
           strcpy(t,"Pediatrician");
        else if(c==2)
           strcpy(t,"Neurology");
        else if(c==3)
           strcpy(t,"Cardiology");
        else if(c==4)
           strcpy(t,"General Physician");
        else if(c==5)
           strcpy(t,"Gynecology");
        else
           printf("invalid choice\n");
        printf("treatment: ");
                scanf("%s",tr);
                printf("\n");
                push(p,s,d,t,tr);
    }
    else
        printf("patient number is incorrect.\n\n");
}

//to display queue of particular department
void see_queue(){
```

```
        int c;
        printf("\nchoose  department: \n1.Pediatric\n2.Neurology\n3.Cardiology\n4.General Physician\n5.Gynecology\nenter choice: ");
   scanf("%d",&c);
   if(c==1)
      display(qp,fp,rp);
   else if(c==2)
      display(qn,fn,rn);
   else if(c==3)
      display(qc,fc,rc);
   else if(c==4)
      display(qg,fg,rg);
   else if(c==5)
      display(qy,fy,ry);
   else
      printf("invalid choice\n");
}

//KRUPA AND MISARI

void enqueue(int element,int q[],int* front,int* rear)
{
   if (*rear == 9)
      printf("Queue Overflow\n");
   else
   {
      if (*front == - 1)
         *front = 0;
      *rear=*rear+1;
      q[*rear] = element;
      printf("%d %d",*front,*rear);
```

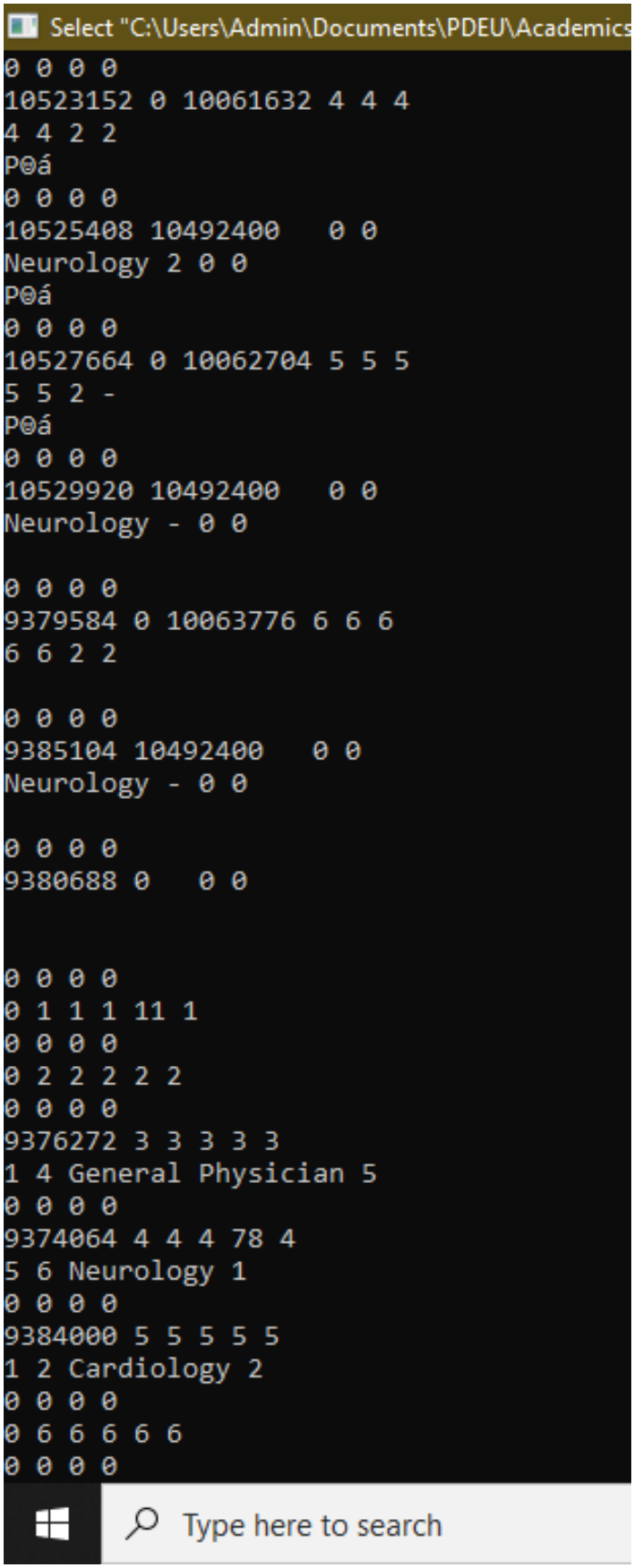
```
    }
}

void dequeue(int q[],int* front,int* rear)
{
    if((*front==-1)||(*front>*rear))
        printf("Queue Underflow \n");
    else
    {
        printf("This patient has been removed from the queue successfully: %d\n", q[*front]);
        *front=*front+1;
    }
}

void display(int q[],int front,int rear)
{
    int i;
    if (front == - 1)
        printf("Queue underflow\n");
    else
    {
        printf("The elements of the queue are:\n");
        for (i = front; i <= rear; i++)
            printf("%d\n",q[i]);
    }
}

void add_queue()
{
    int num;
```

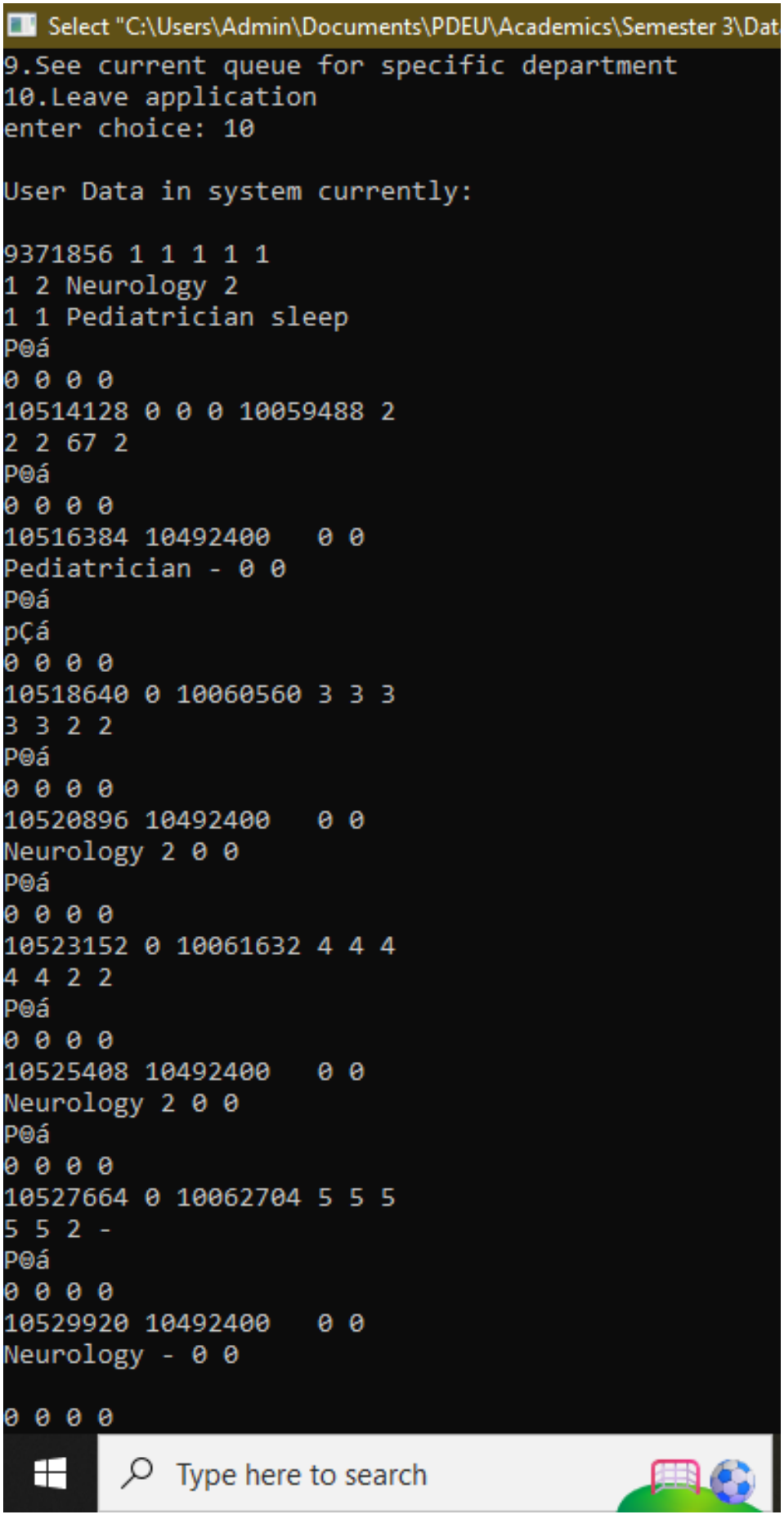
```
        printf("Enter patient no.: ");
        scanf("%d",&num);
        struct node* p;
        p=traverse(num);
        if(p!=NULL)
   {
      int c;
      printf("\nchoose  department:\n1.Pediatric\n2.Neurology\n3.Cardiology\n4.General
Physician\n5.Gynecology\nenter choice: ");
      scanf("%d",&c);
      if(c==1)
         enqueue(p->data.p_no,qp,&fp,&rp);
      else if(c==2)
         enqueue(p->data.p_no,qn,&fn,&rn);
      else if(c==3)
         enqueue(p->data.p_no,qc,&fc,&rc);
      else if(c==4)
         enqueue(p->data.p_no,qg,&fg,&rg);
      else if(c==5)
         enqueue(p->data.p_no,qy,&fy,&ry);
      else
         printf("invalid choice\n");
   }
   else
      printf("patient number is incorrect.\n\n");
}

void out_queue(){
        int num;
        printf("Enter patient no.: ");
        scanf("%d",&num);
```

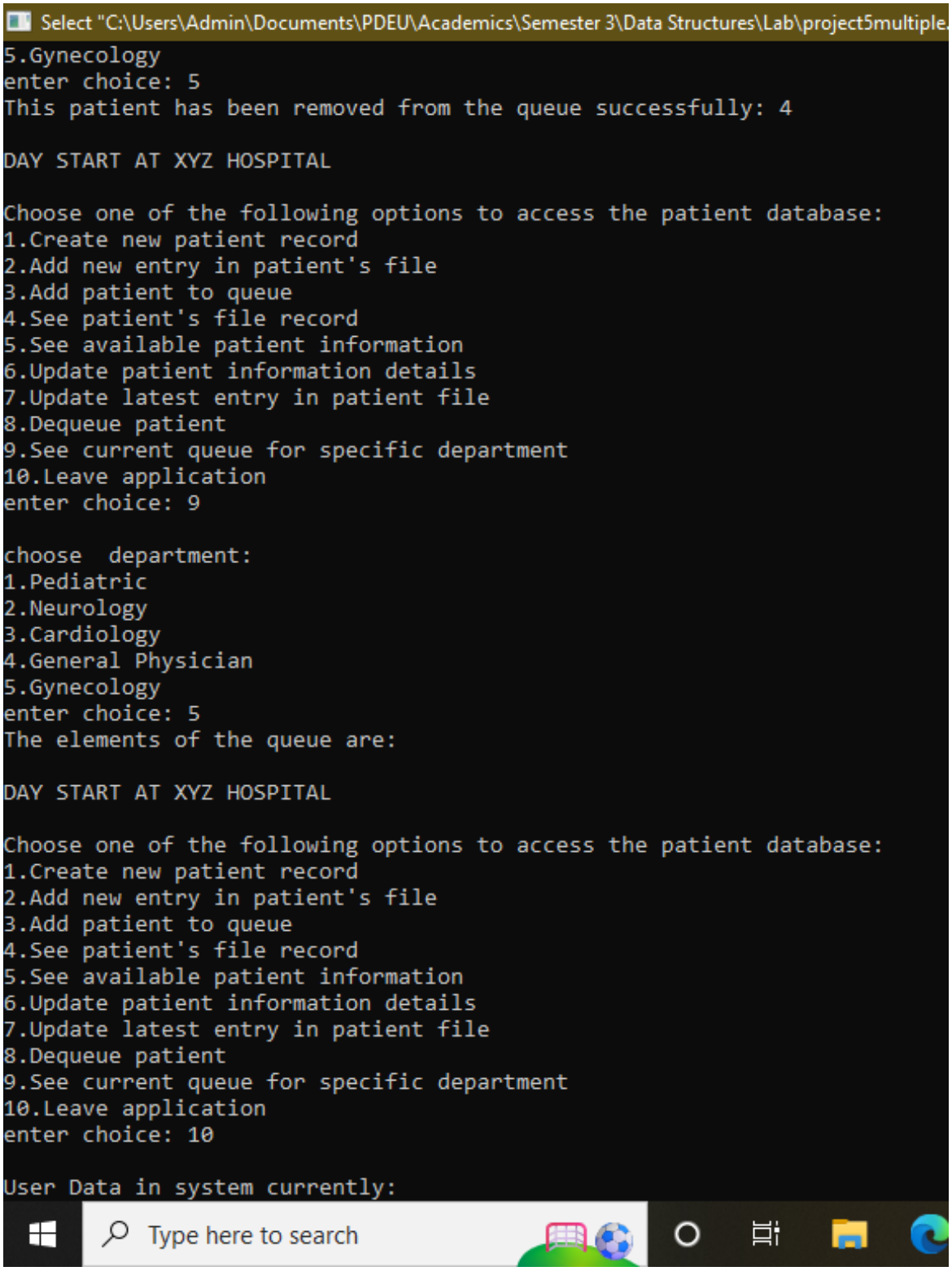
```
        struct node* p;
        p=traverse(num);
        if(p!=NULL)
   {
      int c;
      printf("\nchoose  department: \n1.Pediatric\n2.Neurology\n3.Cardiology\n4.General Physician\n5.Gynecology\nenter choice: ");
      scanf("%d",&c);
      if(c==1)
         dequeue(qp,&fp,&rp);
      else if(c==2)
         dequeue(qn,&fn,&rn);
      else if(c==3)
         dequeue(qc,&fc,&rc);
      else if(c==4)
         dequeue(qg,&fg,&rg);
      else if(c==5)
         dequeue(qy,&fy,&ry);
      else
         printf("invalid choice\n");
         }
}

//VRUNDA

//function to display file stack
void see_stack()
{
   int num,c;
        printf("Enter patient no.: ");
        scanf("%d",&num);
```

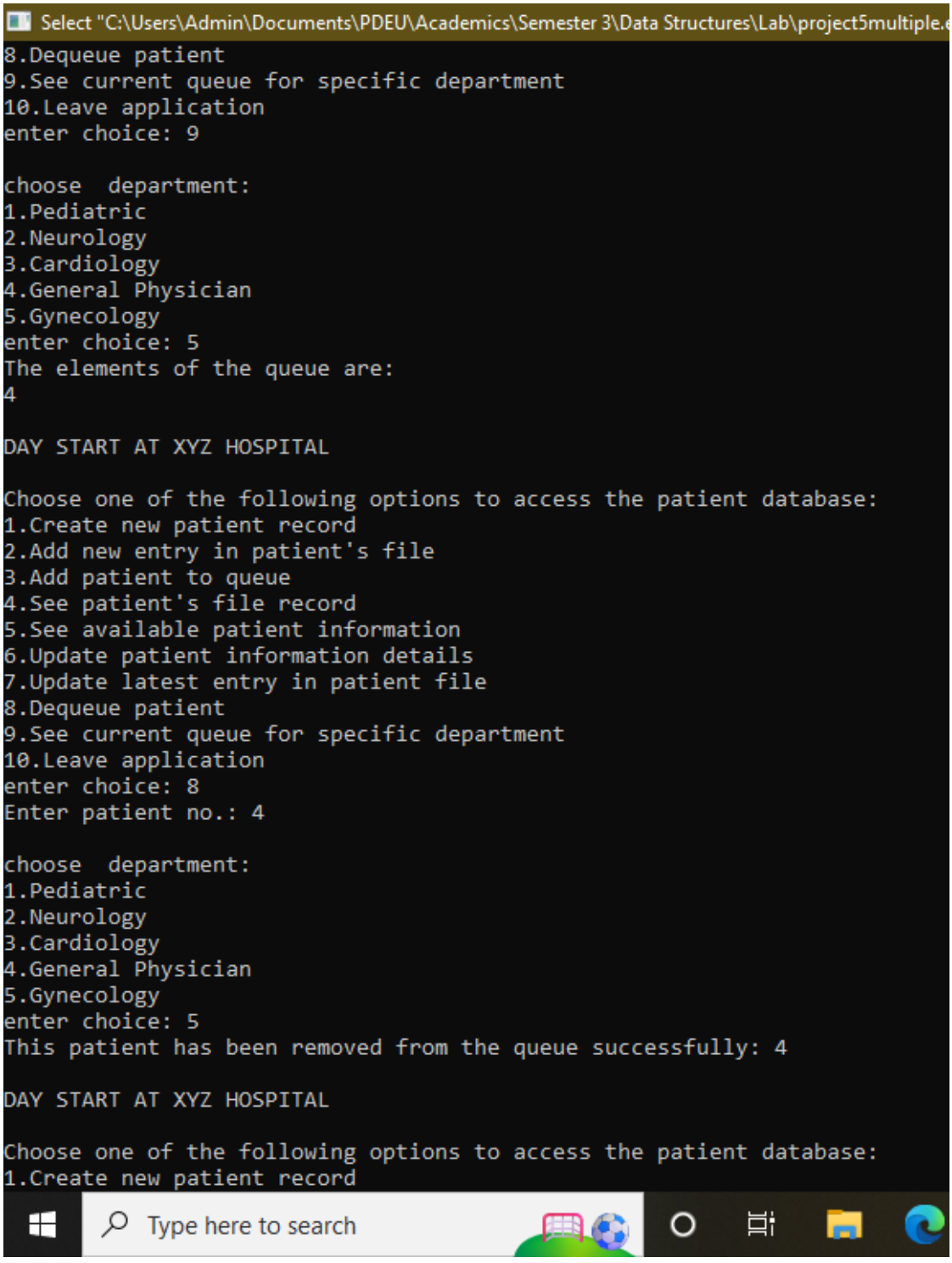
```
        struct node* p;
        p=traverse(num);
        if(p!=NULL)
   {
      struct node_s* top1;
      top1 = p->data.start;
      while (top1 != NULL)
      {
        printf("symptoms: %s\ndiagnosis: %s\ndepartment: %s\ntreatment: %s\n\n",top1-
>data.symptoms,top1->data.diagnosis,top1->data.type,top1->data.treatment);
        top1 = top1->link;
      }
   }
   else
      printf("patient number is incorrect.\n\n");
}

//function to pop entry from stack
struct node_s* pop(struct node* p1)
{
  if (p1->data.start == NULL)
    return p1->data.start;
   struct node_s* q;
   q=p1->data.start;
   p1->data.start = p1->data.start->link;
   return q;
}
```

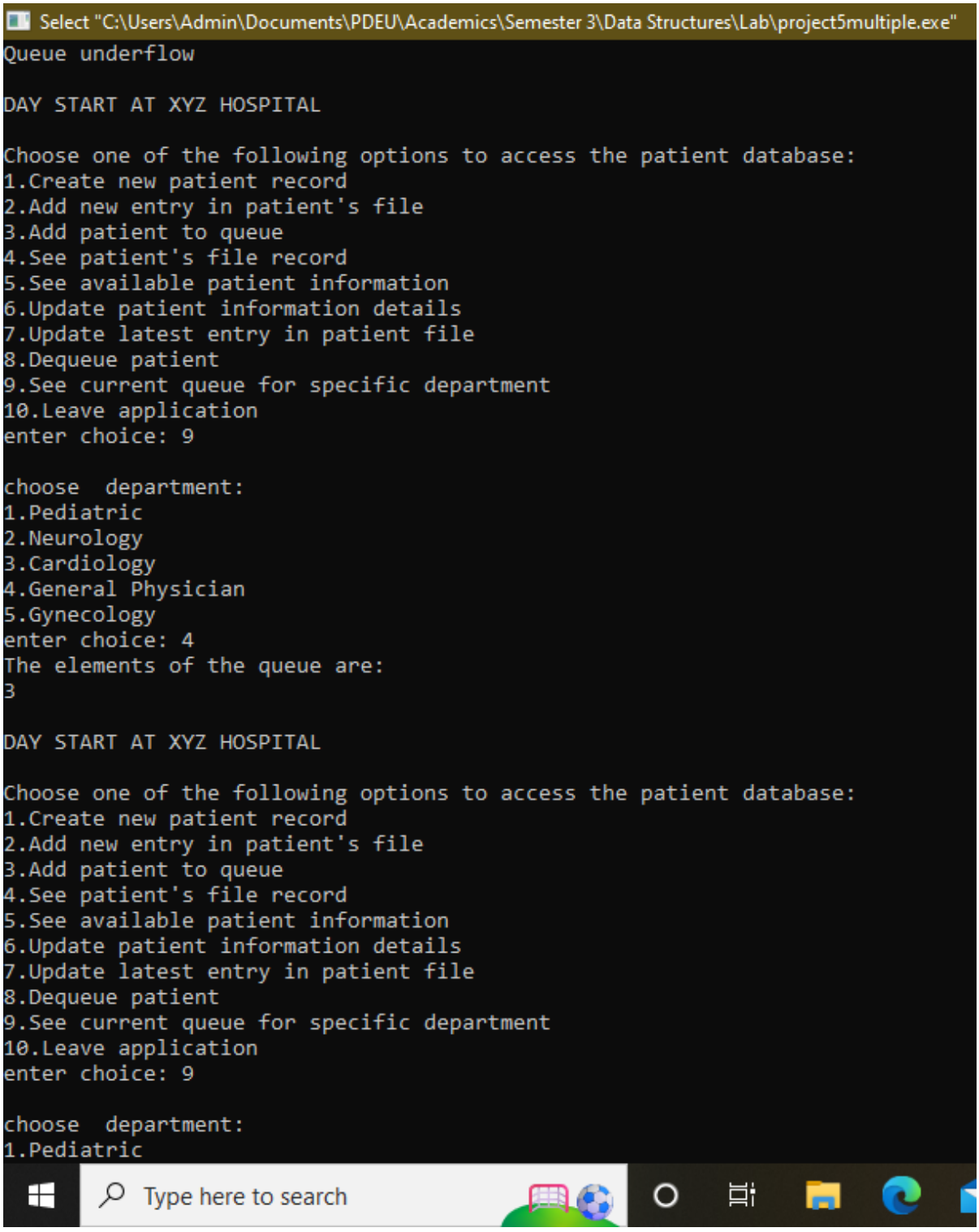
Output:

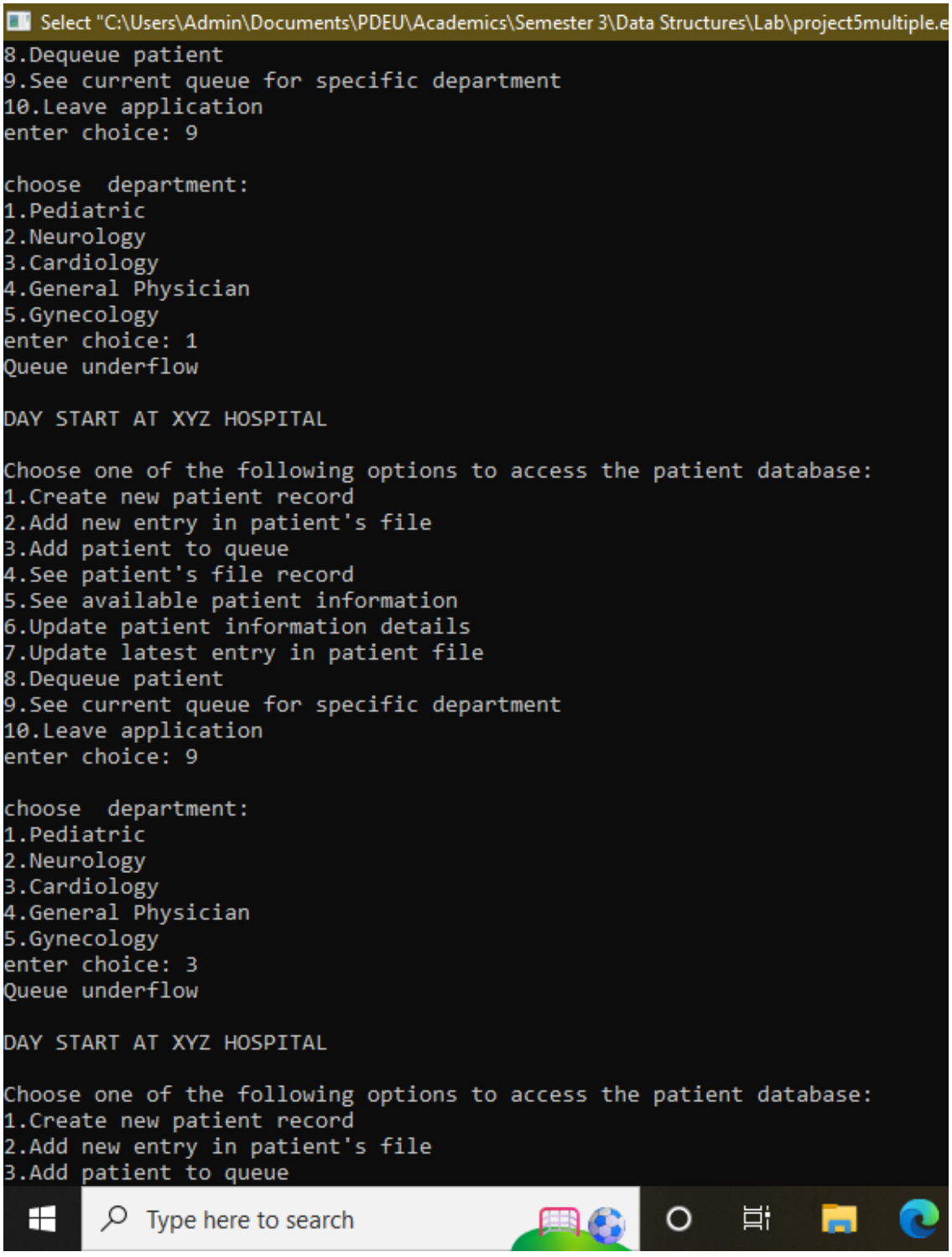

Select "C:\Users\Admin\Documents\PDEU\Academics
0 0 0 0
10523152 0 10061632 4 4 4
4 4 2 2
P@á
0 0 0 0
10525408 10492400 0 0
Neurology 2 0 0
P@á
0 0 0 0
10527664 0 10062704 5 5 5
5 5 2 -
P@á
0 0 0 0
10529920 10492400 0 0
Neurology - 0 0
0 0 0 0
9379584 0 10063776 6 6 6
6 6 2 2
0 0 0 0
9385104 10492400 0 0
Neurology - 0 0
0 0 0 0
9380688 0 0 0
0 0 0 0
0 1 1 1 11 1
0 0 0 0
0 2 2 2 2 2
0 0 0 0
9376272 3 3 3 3 3
1 4 General Physician 5
0 0 0 0
9374064 4 4 4 78 4
5 6 Neurology 1
0 0 0 0
9384000 5 5 5 5 5
1 2 Cardiology 2
0 0 0 0
0 6 6 6 6 6
0 0 0 0
Type here to search

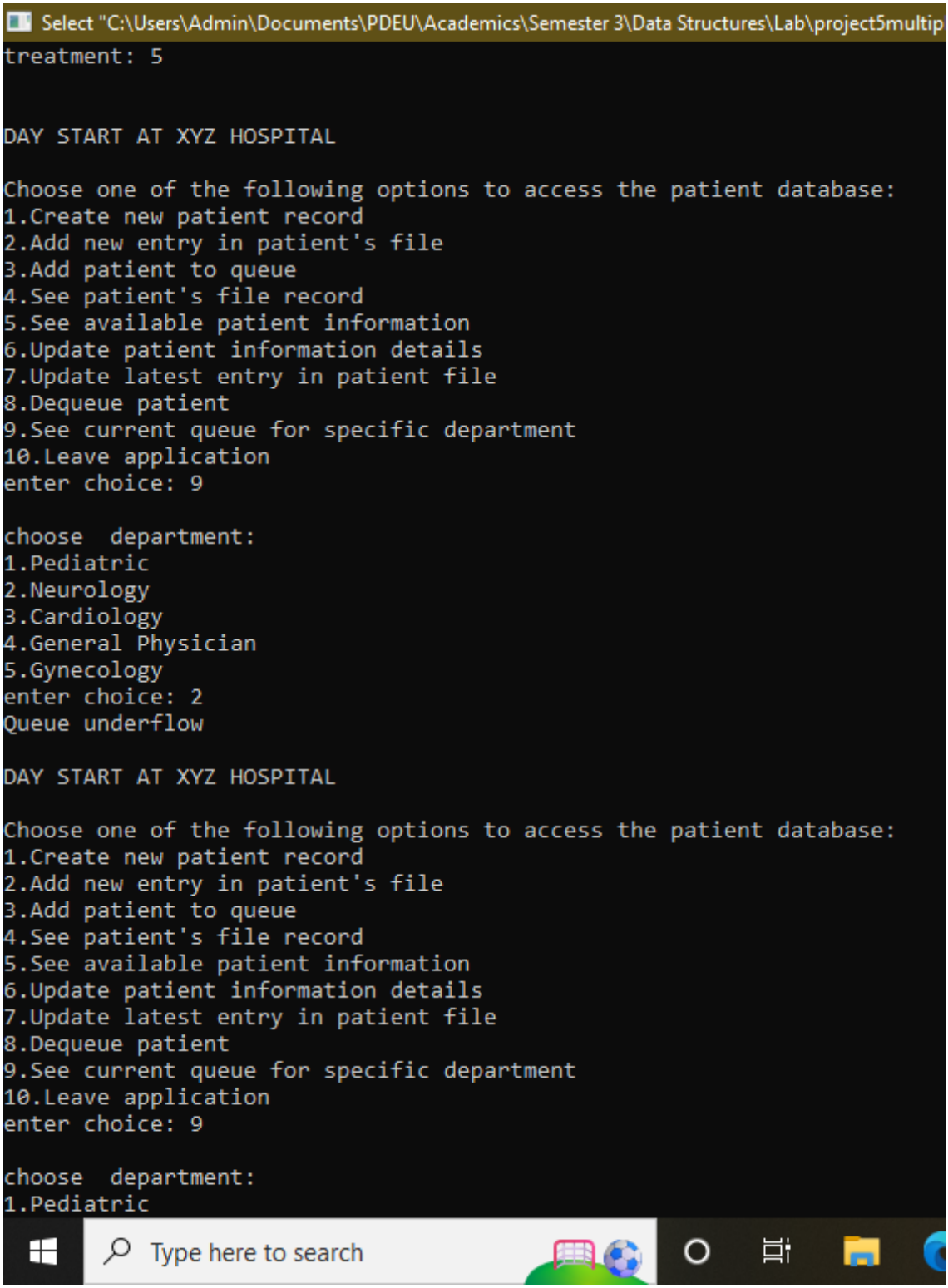

Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Dat
9.See current queue for specific department
10.Leave application
enter choice: 10
User Data in system currently:
9371856 1 1 1 1 1
1 2 Neurology 2
1 1 Pediatrician sleep
P@á
0 0 0 0
10514128 0 0 0 10059488 2
2 2 67 2
P@á
0 0 0 0
10516384 10492400 0 0
Pediatrician - 0 0
P@á
pÇá
0 0 0 0
10518640 0 10060560 3 3 3
3 3 2 2
P@á
0 0 0 0
10520896 10492400 0 0
Neurology 2 0 0
P@á
0 0 0 0
10523152 0 10061632 4 4 4
4 4 2 2
P@á
0 0 0 0
10525408 10492400 0 0
Neurology 2 0 0
P@á
0 0 0 0
10527664 0 10062704 5 5 5
5 5 2 -
P@á
0 0 0 0
10529920 10492400 0 0
Neurology - 0 0
0 0 0 0
Type here to search

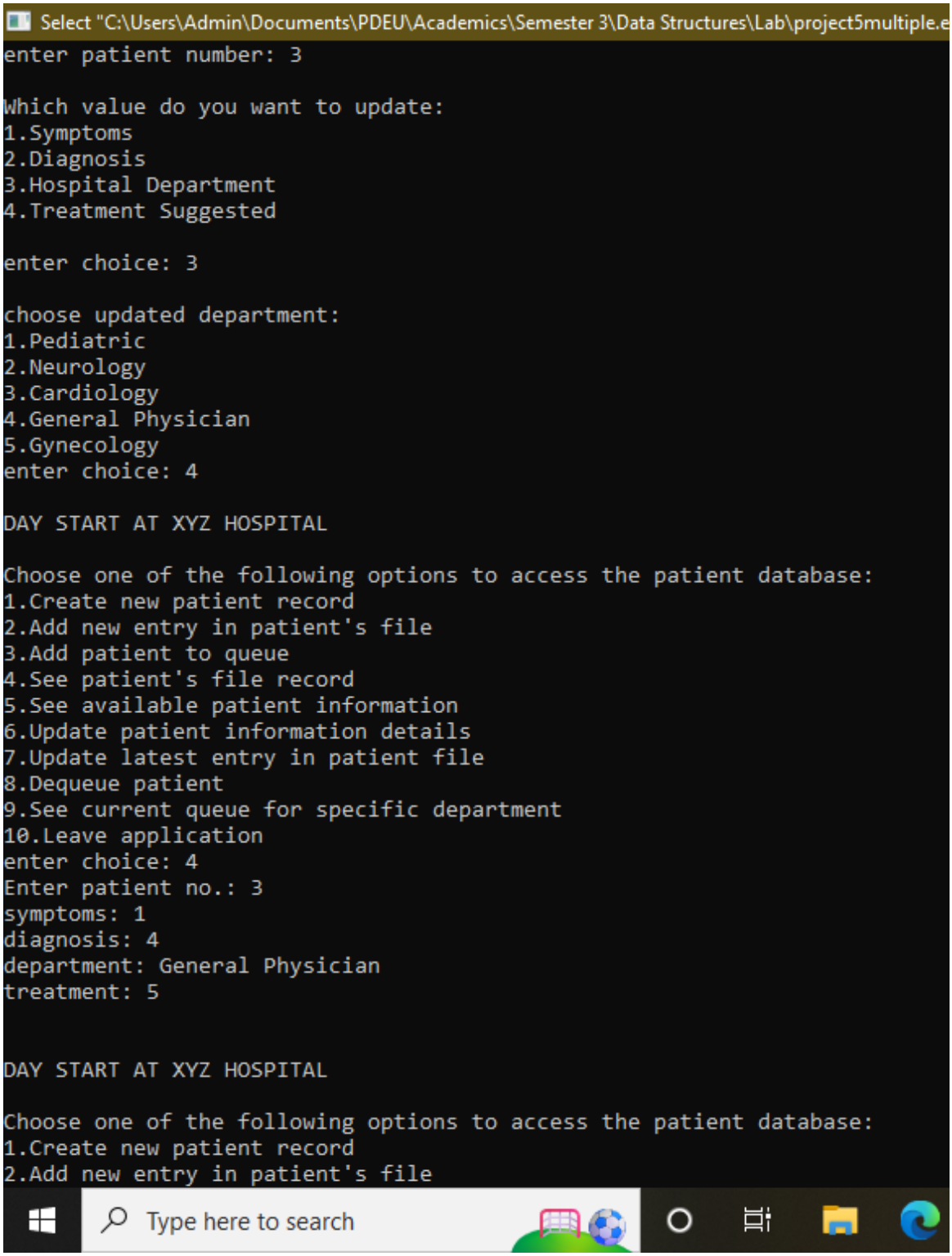

Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multiple
5.Gynecology
enter choice: 5
This patient has been removed from the queue successfully: 4
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 9
choose department:
1.Pediatric
2.Neurology
3.Cardiology
4.General Physician
5.Gynecology
enter choice: 5
The elements of the queue are:
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 10
User Data in system currently:
Type here to search

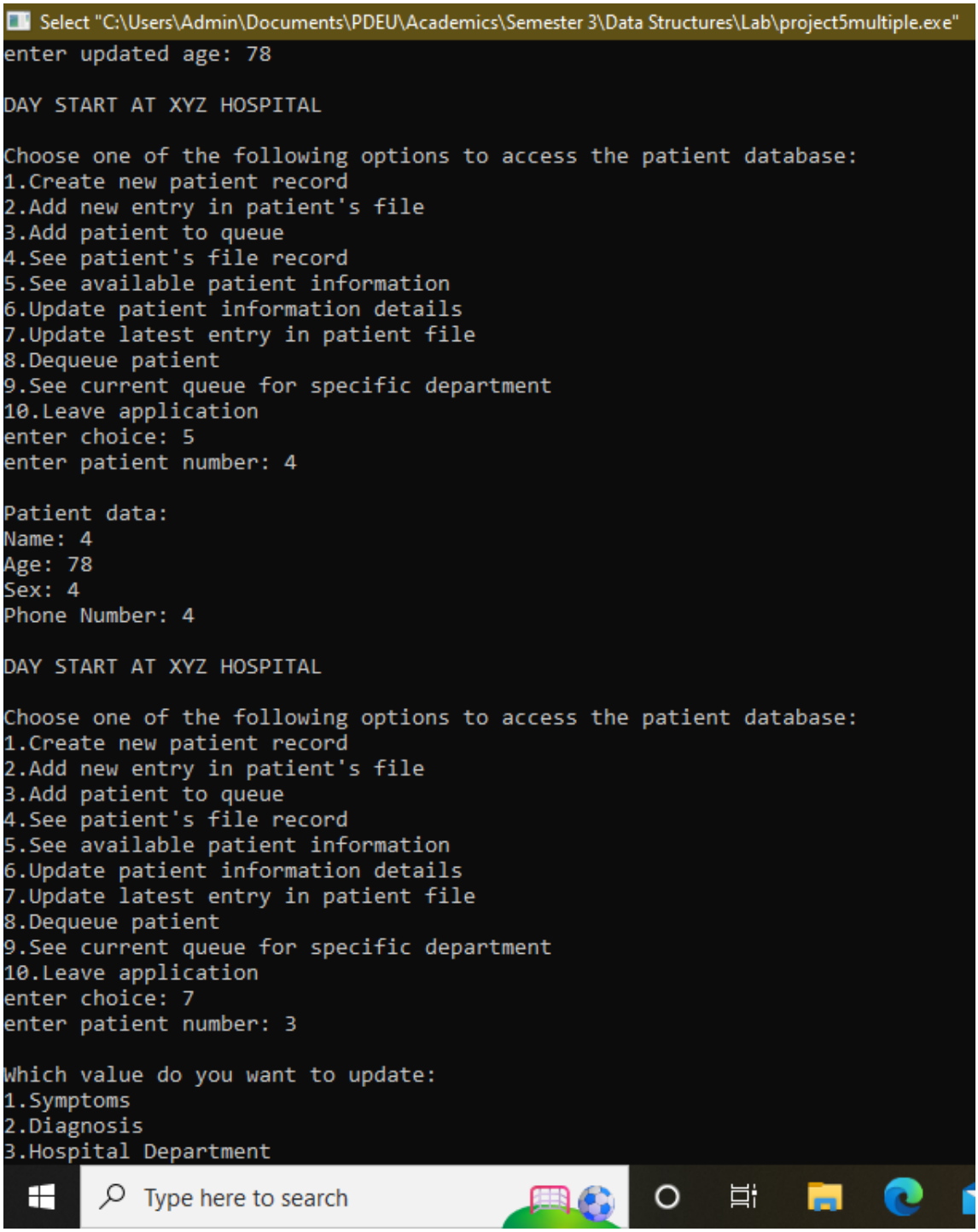

Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multiple.e
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 9
choose department:
1.Pediatric
2.Neurology
3.Cardiology
4.General Physician
5.Gynecology
enter choice: 5
The elements of the queue are:
4
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 8
Enter patient no.: 4
choose department:
1.Pediatric
2.Neurology
3.Cardiology
4.General Physician
5.Gynecology
enter choice: 5
This patient has been removed from the queue successfully: 4
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
Type here to search

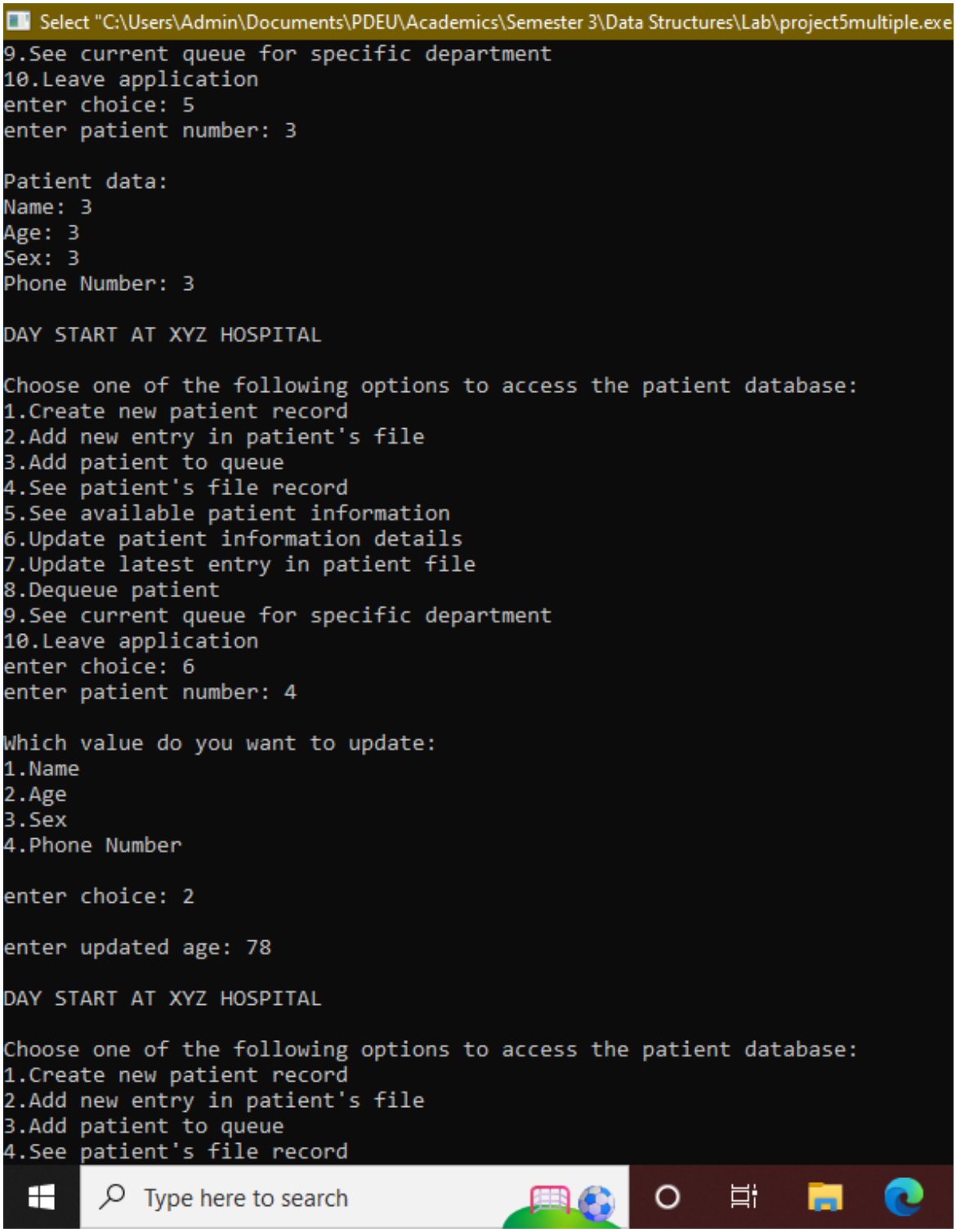

Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multiple.exe"
Queue underflow
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 9
choose department:
1.Pediatric
2.Neurology
3.Cardiology
4.General Physician
5.Gynecology
enter choice: 4
The elements of the queue are:
3
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 9
choose department:
1.Pediatric
Type here to search

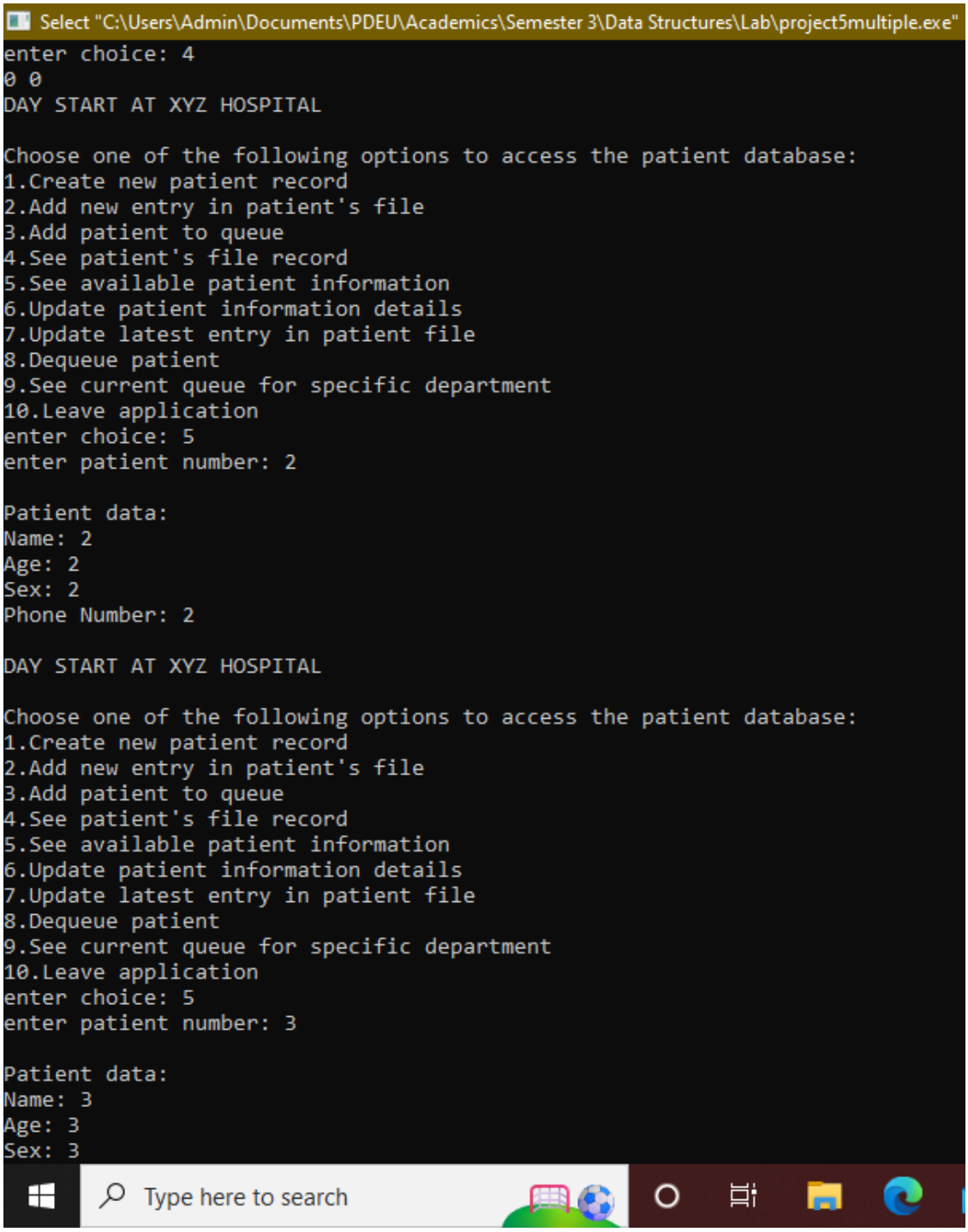

Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multiple.e
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 9
choose department:
1.Pediatric
2.Neurology
3.Cardiology
4.General Physician
5.Gynecology
enter choice: 1
Queue underflow
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 9
choose department:
1.Pediatric
2.Neurology
3.Cardiology
4.General Physician
5.Gynecology
enter choice: 3
Queue underflow
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
Type here to search

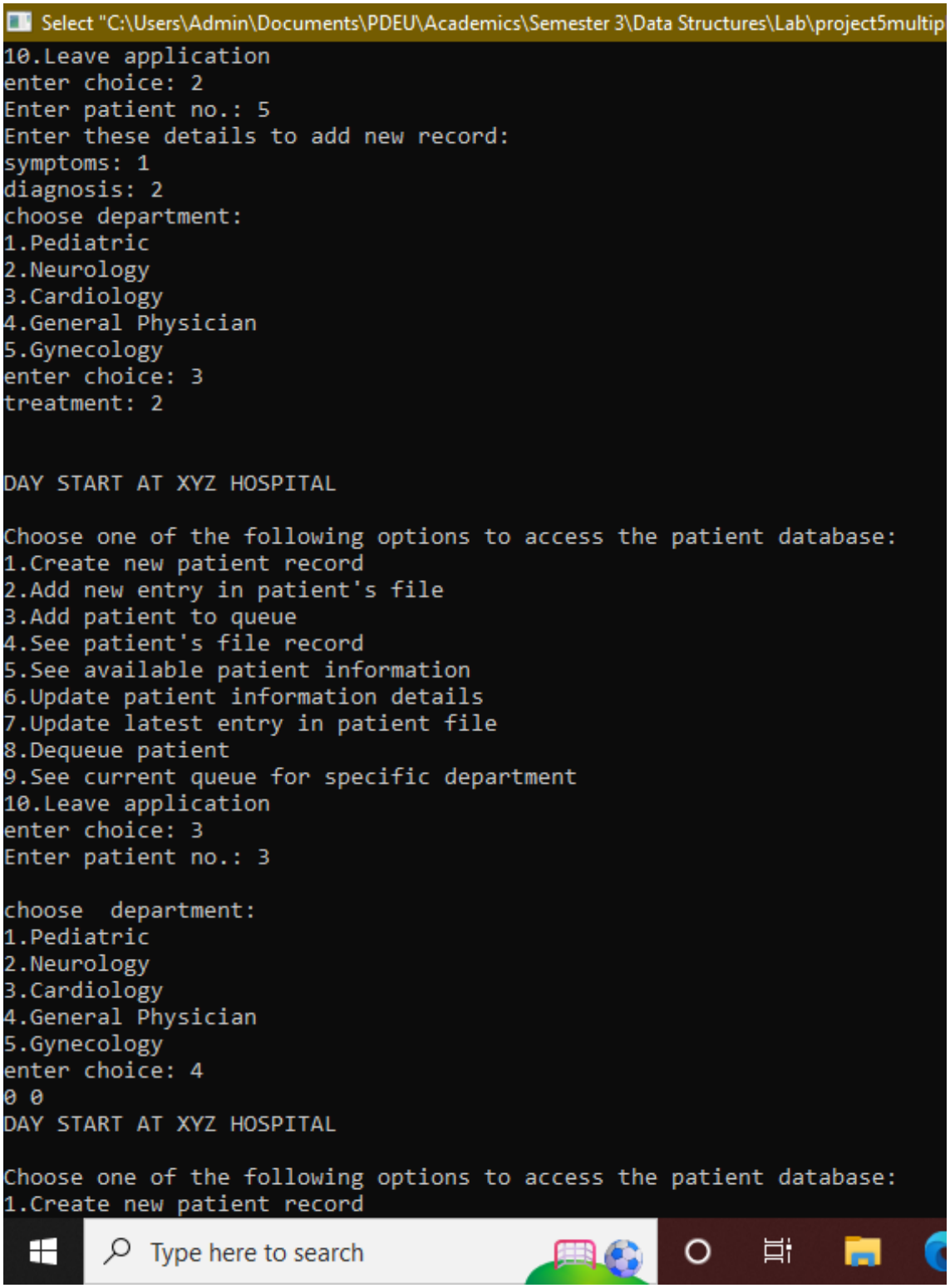

Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multip
treatment: 5
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 9
choose department:
1.Pediatric
2.Neurology
3.Cardiology
4.General Physician
5.Gynecology
enter choice: 2
Queue underflow
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 9
choose department:
1.Pediatric
Type here to search

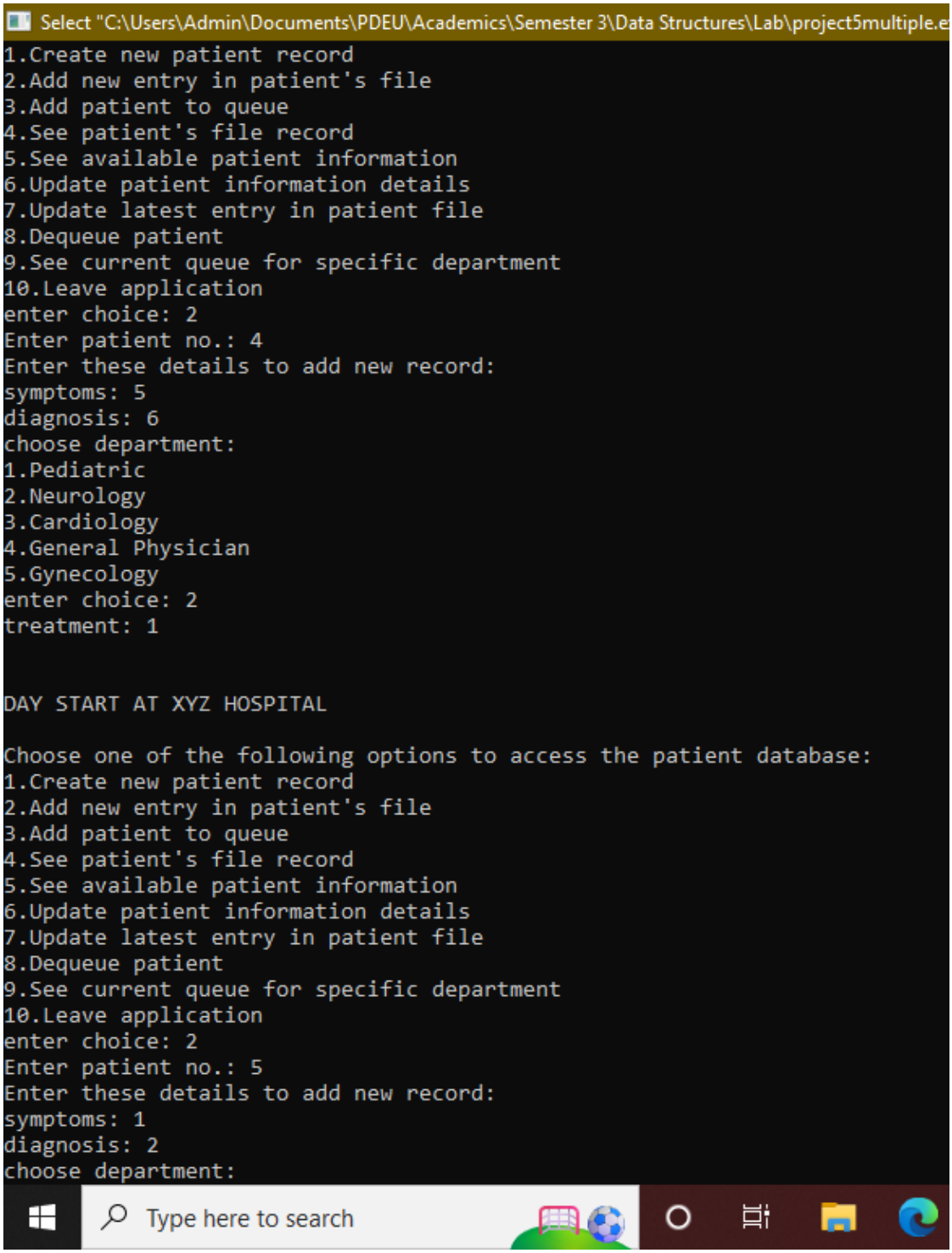

Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multiple.e
enter patient number: 3
Which value do you want to update:
1.Symptoms
2.Diagnosis
3.Hospital Department
4.Treatment Suggested
enter choice: 3
choose updated department:
1.Pediatric
2.Neurology
3.Cardiology
4.General Physician
5.Gynecology
enter choice: 4
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 4
Enter patient no.: 3
symptoms: 1
diagnosis: 4
department: General Physician
treatment: 5
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
Type here to search

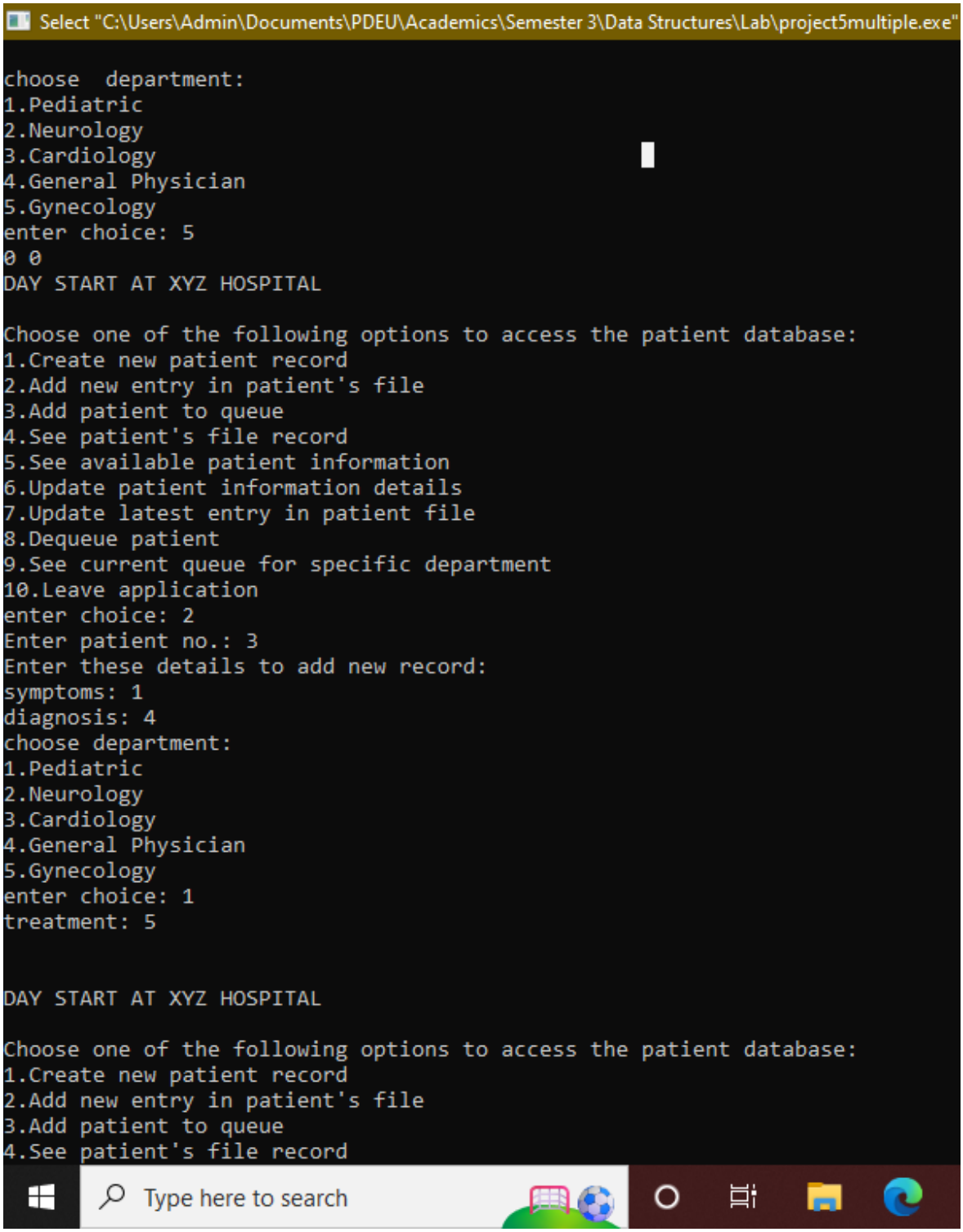

Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multiple.exe"
enter updated age: 78
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 5
enter patient number: 4
Patient data:
Name: 4
Age: 78
Sex: 4
Phone Number: 4
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 7
enter patient number: 3
Which value do you want to update:
1.Symptoms
2.Diagnosis
3.Hospital Department
Type here to search


Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multiple.exe
9.See current queue for specific department
10.Leave application
enter choice: 5
enter patient number: 3
Patient data:
Name: 3
Age: 3
Sex: 3
Phone Number: 3
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 6
enter patient number: 4
Which value do you want to update:
1.Name
2.Age
3.Sex
4.Phone Number
enter choice: 2
enter updated age: 78
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
Type here to search

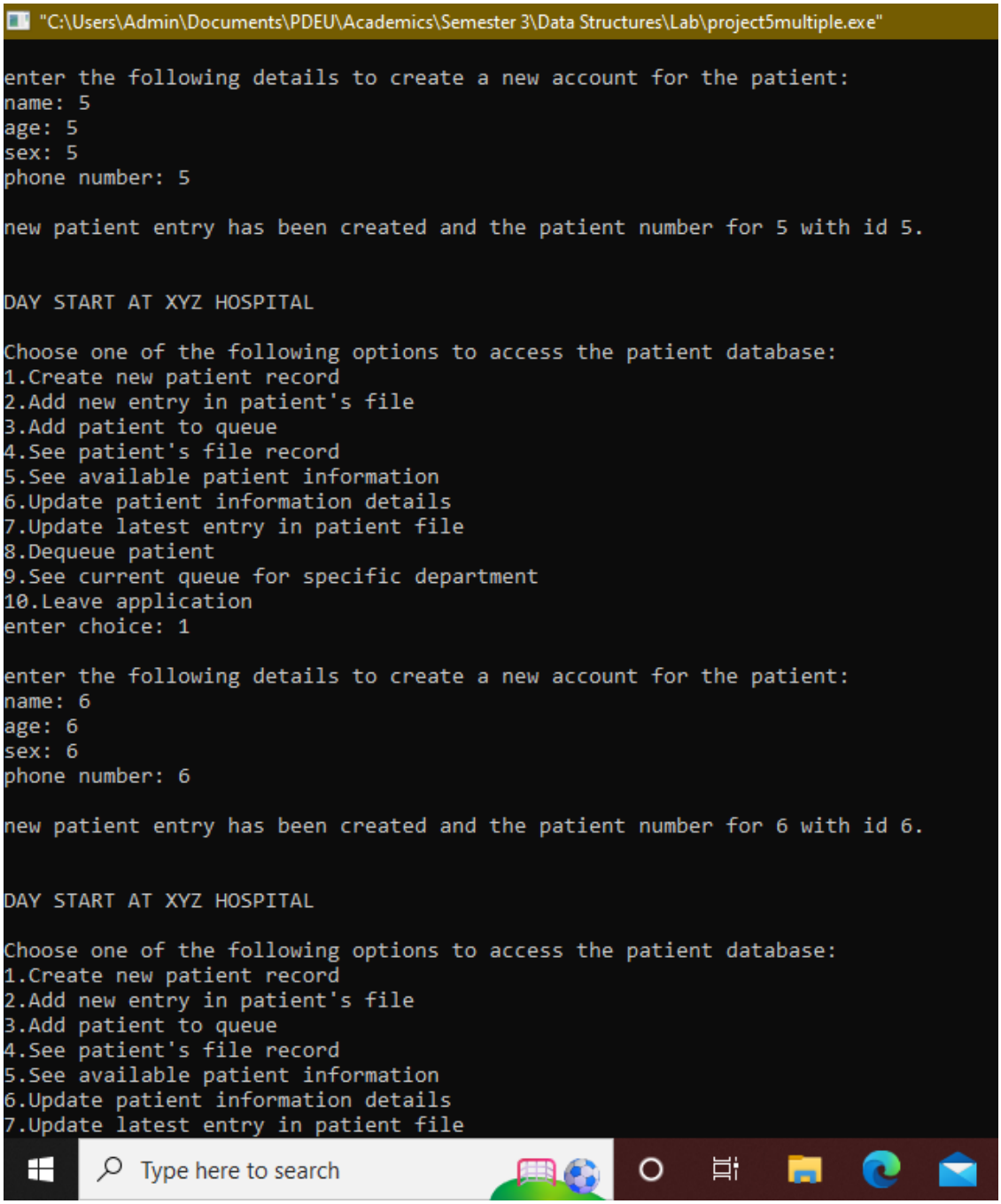

Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multiple.exe"
enter choice: 4
0 0
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 5
enter patient number: 2
Patient data:
Name: 2
Age: 2
Sex: 2
Phone Number: 2
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 5
enter patient number: 3
Patient data:
Name: 3
Age: 3
Sex: 3
Type here to search

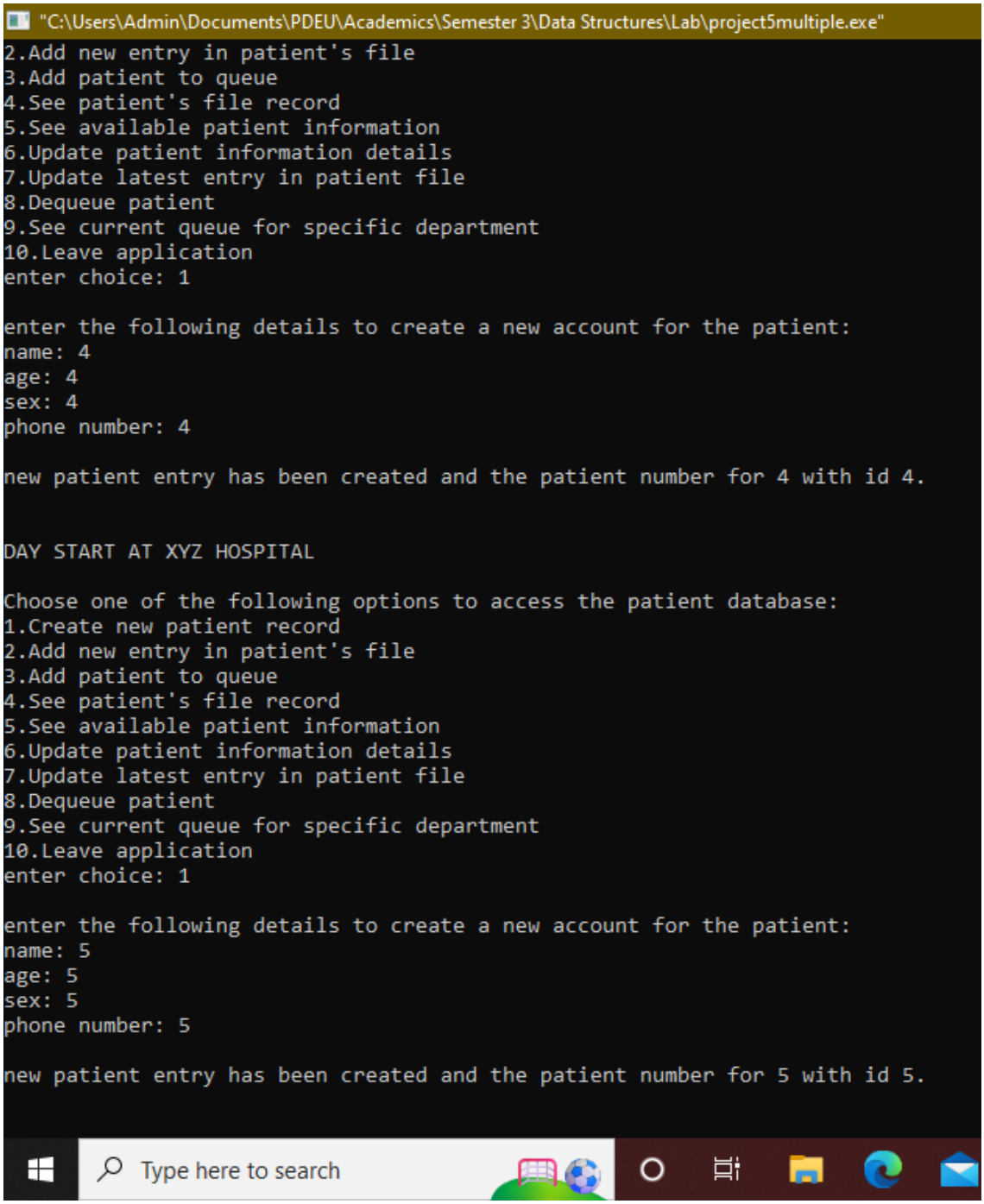

Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multip
10.Leave application
enter choice: 2
Enter patient no.: 5
Enter these details to add new record:
symptoms: 1
diagnosis: 2
choose department:
1.Pediatric
2.Neurology
3.Cardiology
4.General Physician
5.Gynecology
enter choice: 3
treatment: 2
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 3
Enter patient no.: 3
choose department:
1.Pediatric
2.Neurology
3.Cardiology
4.General Physician
5.Gynecology
enter choice: 4
0 0
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
Type here to search

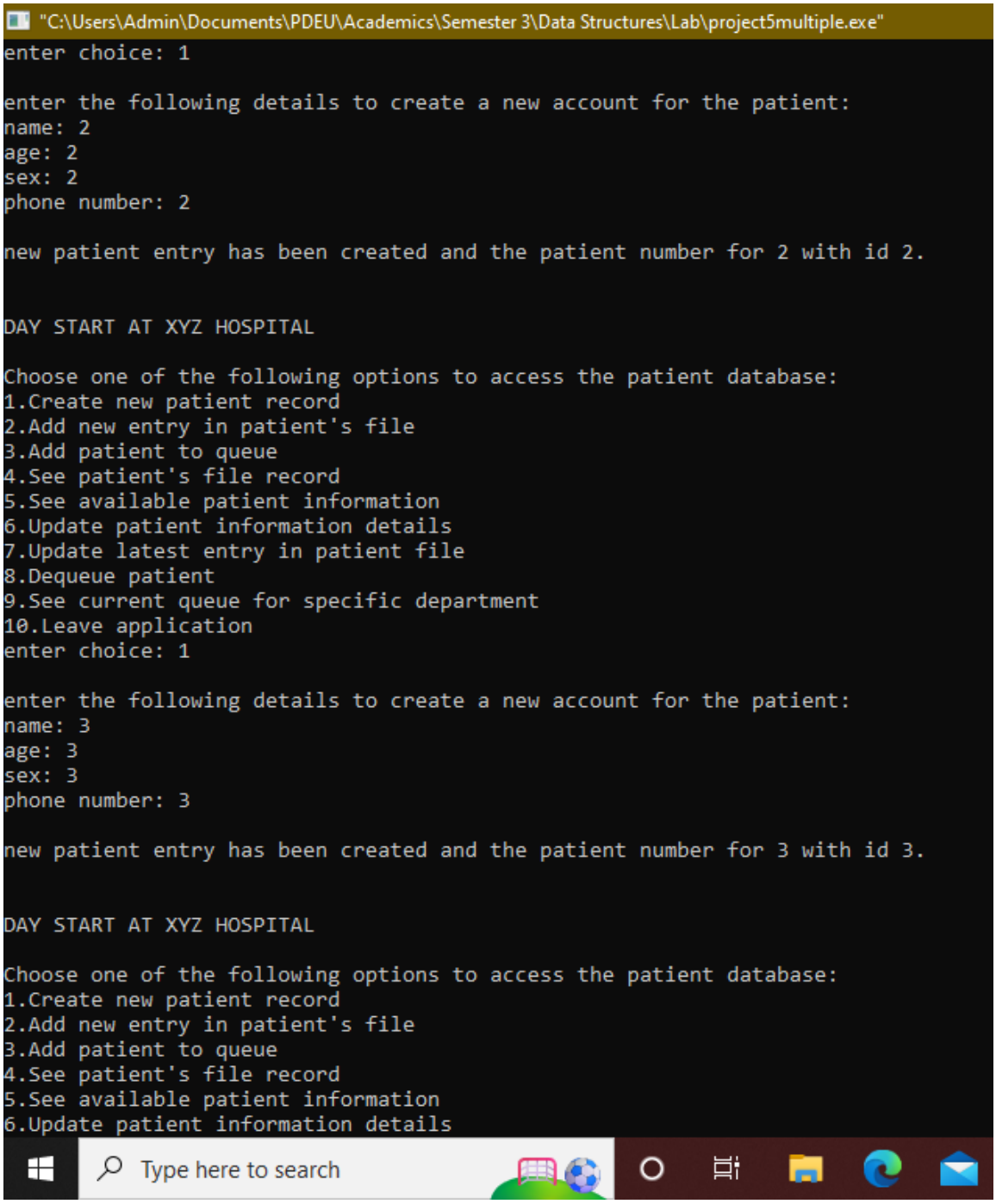

Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multiple.e
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 2
Enter patient no.: 4
Enter these details to add new record:
symptoms: 5
diagnosis: 6
choose department:
1.Pediatric
2.Neurology
3.Cardiology
4.General Physician
5.Gynecology
enter choice: 2
treatment: 1
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 2
Enter patient no.: 5
Enter these details to add new record:
symptoms: 1
diagnosis: 2
choose department:
Type here to search

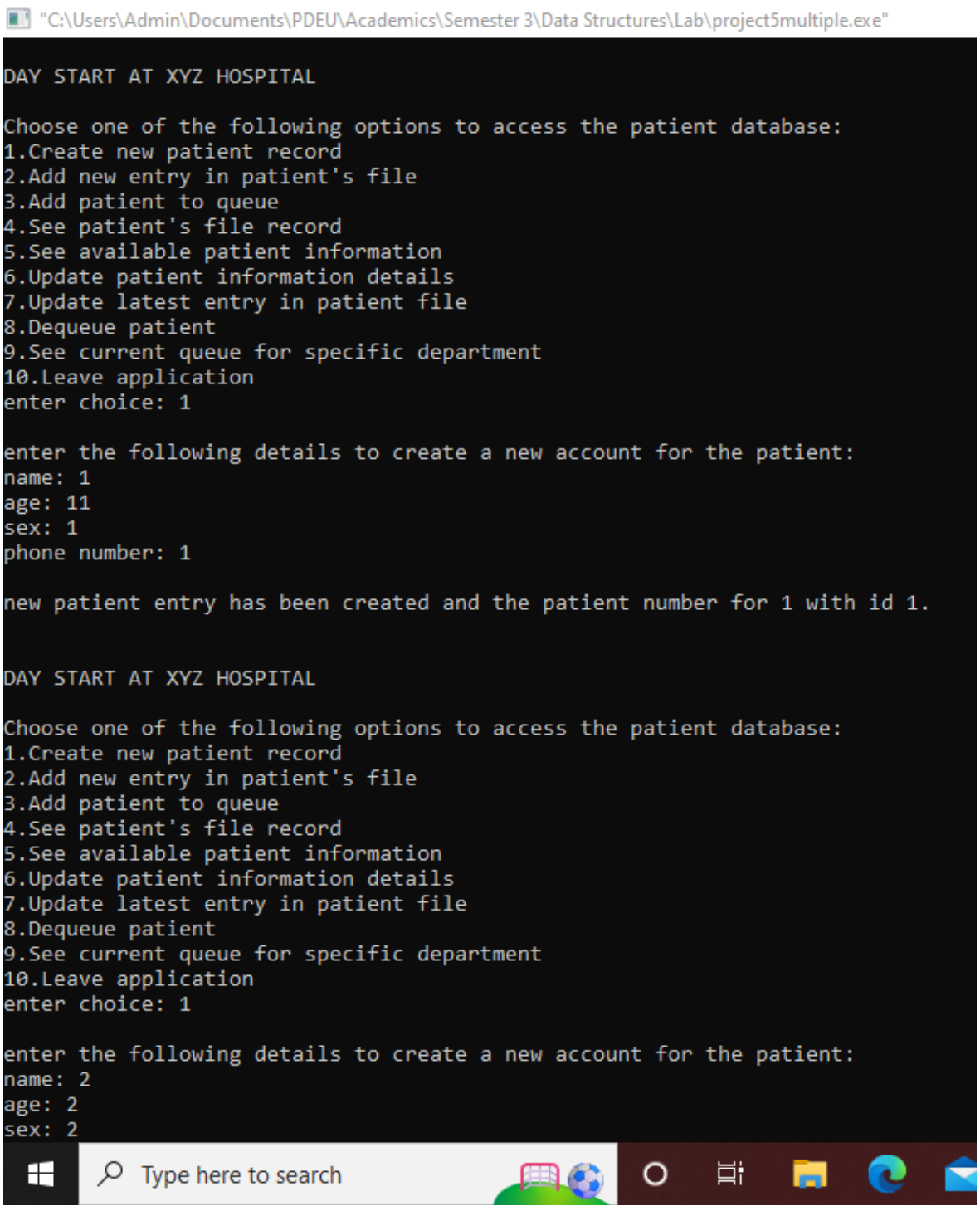

Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multiple.exe"
choose department:
1.Pediatric
2.Neurology
3.Cardiology
4.General Physician
5.Gynecology
enter choice: 5
0 0
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 2
Enter patient no.: 3
Enter these details to add new record:
symptoms: 1
diagnosis: 4
choose department:
1.Pediatric
2.Neurology
3.Cardiology
4.General Physician
5.Gynecology
enter choice: 1
treatment: 5
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
Type here to search

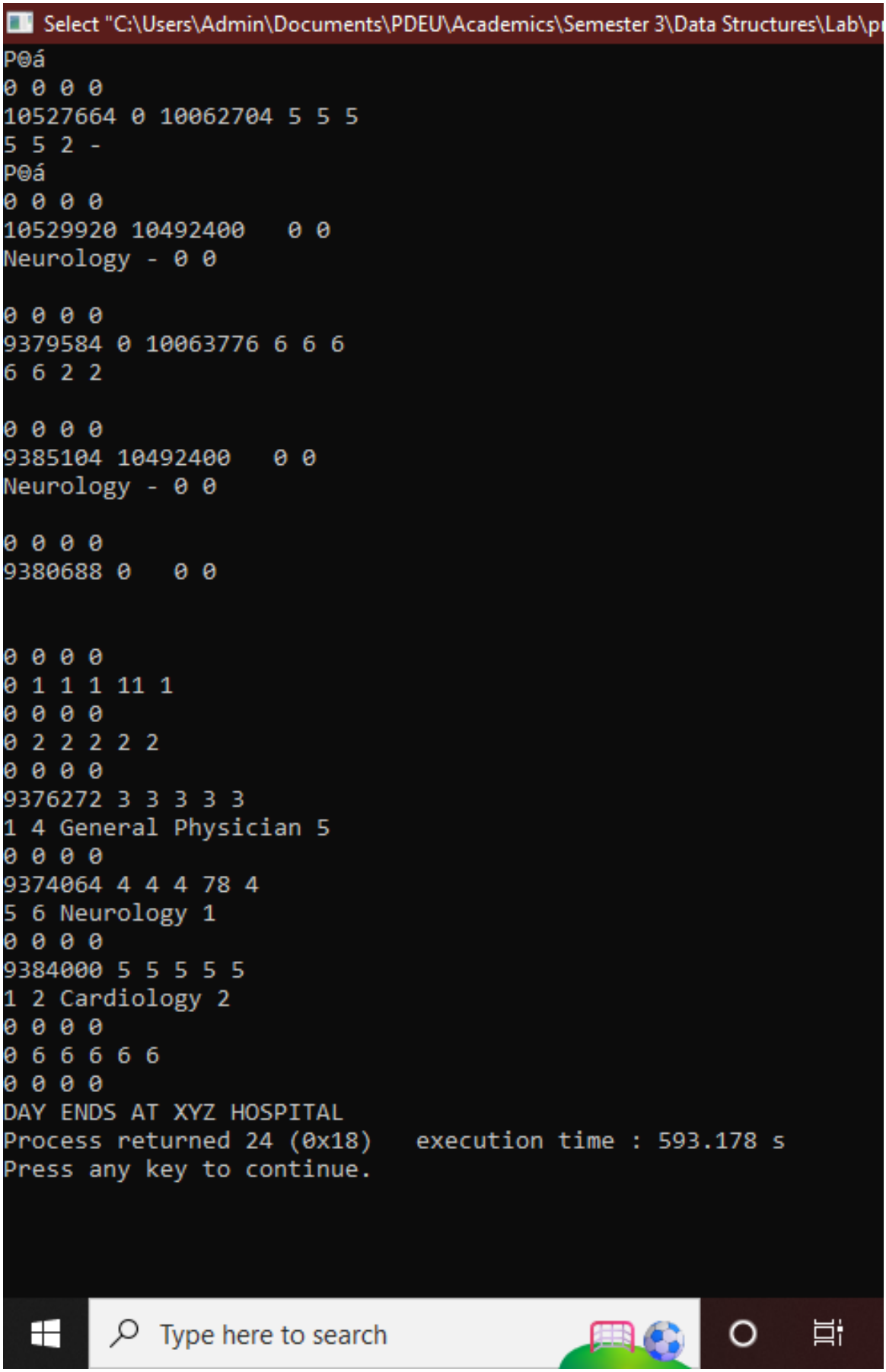

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multiple.exe"
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 2
Enter patient no.: 1
Enter these details to add new record:
symptoms: 1
diagnosis: 2
choose department:
1.Pediatric
2.Neurology
3.Cardiology
4.General Physician
5.Gynecology
enter choice: 2
treatment: 2
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 3
Enter patient no.: 4
choose department:
1.Pediatric
2.Neurology
3.Cardiology
Type here to search


"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multiple.exe"
enter the following details to create a new account for the patient:
name: 5
age: 5
sex: 5
phone number: 5
new patient entry has been created and the patient number for 5 with id 5.
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 1
enter the following details to create a new account for the patient:
name: 6
age: 6
sex: 6
phone number: 6
new patient entry has been created and the patient number for 6 with id 6.
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
Type here to search


"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multiple.exe"
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 1
enter the following details to create a new account for the patient:
name: 4
age: 4
sex: 4
phone number: 4
new patient entry has been created and the patient number for 4 with id 4.
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 1
enter the following details to create a new account for the patient:
name: 5
age: 5
sex: 5
phone number: 5
new patient entry has been created and the patient number for 5 with id 5.
Type here to search


"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multiple.exe"
enter choice: 1
enter the following details to create a new account for the patient:
name: 2
age: 2
sex: 2
phone number: 2
new patient entry has been created and the patient number for 2 with id 2.
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 1
enter the following details to create a new account for the patient:
name: 3
age: 3
sex: 3
phone number: 3
new patient entry has been created and the patient number for 3 with id 3.
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
Type here to search


"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project5multiple.exe"
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 1
enter the following details to create a new account for the patient:
name: 1
age: 11
sex: 1
phone number: 1
new patient entry has been created and the patient number for 1 with id 1.
DAY START AT XYZ HOSPITAL
Choose one of the following options to access the patient database:
1.Create new patient record
2.Add new entry in patient's file
3.Add patient to queue
4.See patient's file record
5.See available patient information
6.Update patient information details
7.Update latest entry in patient file
8.Dequeue patient
9.See current queue for specific department
10.Leave application
enter choice: 1
enter the following details to create a new account for the patient:
name: 2
age: 2
sex: 2
Type here to search


Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\p
P@á
0 0 0 0
10527664 0 10062704 5 5 5
5 5 2 -
P@á
0 0 0 0
10529920 10492400 0 0
Neurology - 0 0
0 0 0 0
9379584 0 10063776 6 6 6
6 6 2 2
0 0 0 0
9385104 10492400 0 0
Neurology - 0 0
0 0 0 0
9380688 0 0 0
0 0 0 0
0 1 1 1 11 1
0 0 0 0
0 2 2 2 2 2
0 0 0 0
9376272 3 3 3 3 3
1 4 General Physician 5
0 0 0 0
9374064 4 4 4 78 4
5 6 Neurology 1
0 0 0 0
9384000 5 5 5 5 5
1 2 Cardiology 2
0 0 0 0
0 6 6 6 6 6
0 0 0 0
DAY ENDS AT XYZ HOSPITAL
Process returned 24 (0x18) execution time : 593.178 s
Press any key to continue.
Type here to search

Data in File:

9371856 1 1 1 1 1

1 2 Neurology 2

1 1 Pediatrician sleep

P

0 0 0 0

10514128 0 0 0 10059488 2

2 2 67 2

P

0 0 0 0

10516384 10492400 0 0

Pediatrician - 0 0

P

p€

0 0 0 0

10518640 0 10060560 3 3 3

3 3 2 2

P

0 0 0 0

10520896 10492400 0 0

Neurology 2 0 0

P

0 0 0 0

10523152 0 10061632 4 4 4

4 4 2 2

P

0 0 0 0

10525408 10492400 0 0

Neurology 2 0 0

P

0 0 0 0

10527664 0 10062704 5 5 5

5 5 2 -

P

0 0 0 0

10529920 10492400 0 0

Neurology - 0 0

0 0 0 0

9379584 0 10063776 6 6 6

6 6 2 2

0 0 0 0

9385104 10492400 0 0

Neurology - 0 0

0 0 0 0

9380688 0 0 0

0 0 0 0

0 1 1 1 11 1

0 0 0 0

0 2 2 2 2 2

0 0 0 0

9376272 3 3 3 3 3

1 4 General Physician 5

0 0 0 0

9374064 4 4 4 78 4

5 6 Neurology 1

0 0 0 0

9384000 5 5 5 5 5

1 2 Cardiology 2

0 0 0 0

0 6 6 6 6 6

0 0 0 0

Project 6: Tree

FACTORIZATION TREE AND LCM, HCF CALCULATION USING LEAF NODES

Code:

```
//task - create a program that uses tree with real-life applications
//PRIME FACTORIZATION TREE

#include<stdio.h>
#include<math.h>
#include<stdlib.h>

//REHA

//btnode structure
struct btnode
{
   struct btnode* left;
   int data;
   struct btnode* right;
};

//declaring 'root' in b.tree as global
struct btnode* root;

//function to create node in binary tree
struct btnode* new_node(int input,struct btnode* address)
{
   address->data=input;
   address->right=NULL;
```

```
    address->left=NULL;
}

//function to display the prime factor tree
void display()
{
    //declaring and defining variables
    struct btnode* p;
    p=root;
    int c=1;  //used to alternate between left and right sub trees as that is how the tree was created
    //printing the root node if it is not null
    if(p!=NULL)
    {
        printf("\t\t%d\n\t       / \\\n\t      ",p->data);
        //printing the node values until the last leaf node
        while(p->left!=NULL)
        {
            printf("%d   %d\n",p->left->data,p->right->data);
            //going to the right sub-tree
            if(c==1)
            {
                p=p->right;
                c=0;
                printf("\t        / \\\n\t\t");
            }
            //going to the left sub-tree
            else
            {
                p=p->left;
                c=1;
```

```
            printf("\t       / \\\n\t      ");
        }
    }
  }
  else
    printf("\nERROR: Can't display an empty tree!!\n");
}

//function to create the prime factors tree
void create(int number)
{
  //declaring and defining required variables
  root=(struct btnode*)malloc(sizeof(struct btnode));
  root->data=number;
  root->left=NULL;
  root->right=NULL;
  struct btnode* p;
  p=root;
  int c=number,i=2,j,x,a=1;
  //creating the nodes in the tree except for the root
  while(i<=sqrt(c))
  {
    //checking if i is a factor of c or not
    if(c%i==0)
    {
      x=1;
      //checking if i is a prime number or not
      for(j=2;j<=sqrt(i);j++)
      {
        if(i%j==0)
```

```
            {
                x=0;
                break;
            }
        }
        //if i is prime then adding it to appropriate node in tree and updating p to point to the other child node
        if(x==1)
        {
            //adding i to left of p if p is right of it's parent or if it is the main root and c/i to the right
            if(a==1)
            {
                p->left=(struct btnode*)malloc(sizeof(struct btnode));
                new_node(i,p->left);
                p->right=(struct btnode*)malloc(sizeof(struct btnode));
                new_node(c/i,p->right);
                p=p->right;
                c/=i;
                a=0;
            }
            //adding i to right of p if p is left of it's parent and c/i to the left
            else
            {
                p->left=(struct btnode*)malloc(sizeof(struct btnode));
                new_node(c/i,p->left);
                p->right=(struct btnode*)malloc(sizeof(struct btnode));
                new_node(i,p->right);
                p=p->left;
                c/=i;
                a=1;
```

```
            }
            i=2;   //restarting the loop for new c
        }
        //checking for another i as this one is not prime
        else
            i+=1;
    }
    //checking for another i as this one isn't a factor of c
    else
        i+=1;
  }
}

//VRUNDA

void findHCF(int a[],int b[])
{
  int result=1,i;
  for(int i=0;i<1000;i++)
  {
    if(a[i]==0)
        break;
    for(int j=0;j<1000;j++)
    {
        if(b[j]==0)
            break;
        if(a[i]==b[j])
        {
            result*=a[i];
            b[j]=1;
```

```
            break;
        }
    }
  }
  printf("%d",result);
}

//REHA

void findLCM(int ap[],int bp[])
{
  int i,j,lcm=1,n1=0,n2=0;
  for(i=0;i<1000;i++)
  {
    if (ap[i]==0)
      break;
    n1+=1;
  }
  for(i=0;i<1000;i++)
  {
    if (bp[i]==0)
      break;
    n2+=1;
  }
  for(i=0;i<n1;i++)
  {
    for(j=0;j<n2;j++)
    {
      if(ap[i]==bp[j])
      {
```

```
            bp[j]=1;
            break;
        }
    }
    lcm*=ap[i];
  }
  for(i=0;i<n2;i++)
  {
    lcm*=bp[i];
  }
  printf("%d",lcm);
}

//KRUPA AND REHA

//globally declaring array to store prime factors
int arr[1000];

//function to find the leafnodes and add them to the array of prime factors
int* leafnode(struct btnode* node)
{
  int i;
  if(node == NULL)
    return arr;
  if(node->left == NULL && node->right==NULL)
  {
    printf("%d, ",node->data);
    for(i=0;i<1000;i++)
    {
      if (arr[i]==0)
```

```
            {
                arr[i]=node->data;
                break;
            }
        }
        return arr;
    }
    else
    {
        int *ar = leafnode(node->left);
        int *br = leafnode(node->right);
    }
    return arr;
}

//TANYA

void main()
 {
          int choice,a,b,c,f=1,i,arp[1000],brp[1000];
          int *ap,*bp;
 while(f==1)
    {
        for(i=0;i<100;i++)
        {
            arr[i]=0;
            arp[i]=0;
            brp[i]=0;
        }
        printf("Choose one of the following:\n\n1.display prime factorization tree\n2.find
LCM\n3.find HCF\n4.Leave application\n\nEnter choice: ");
```

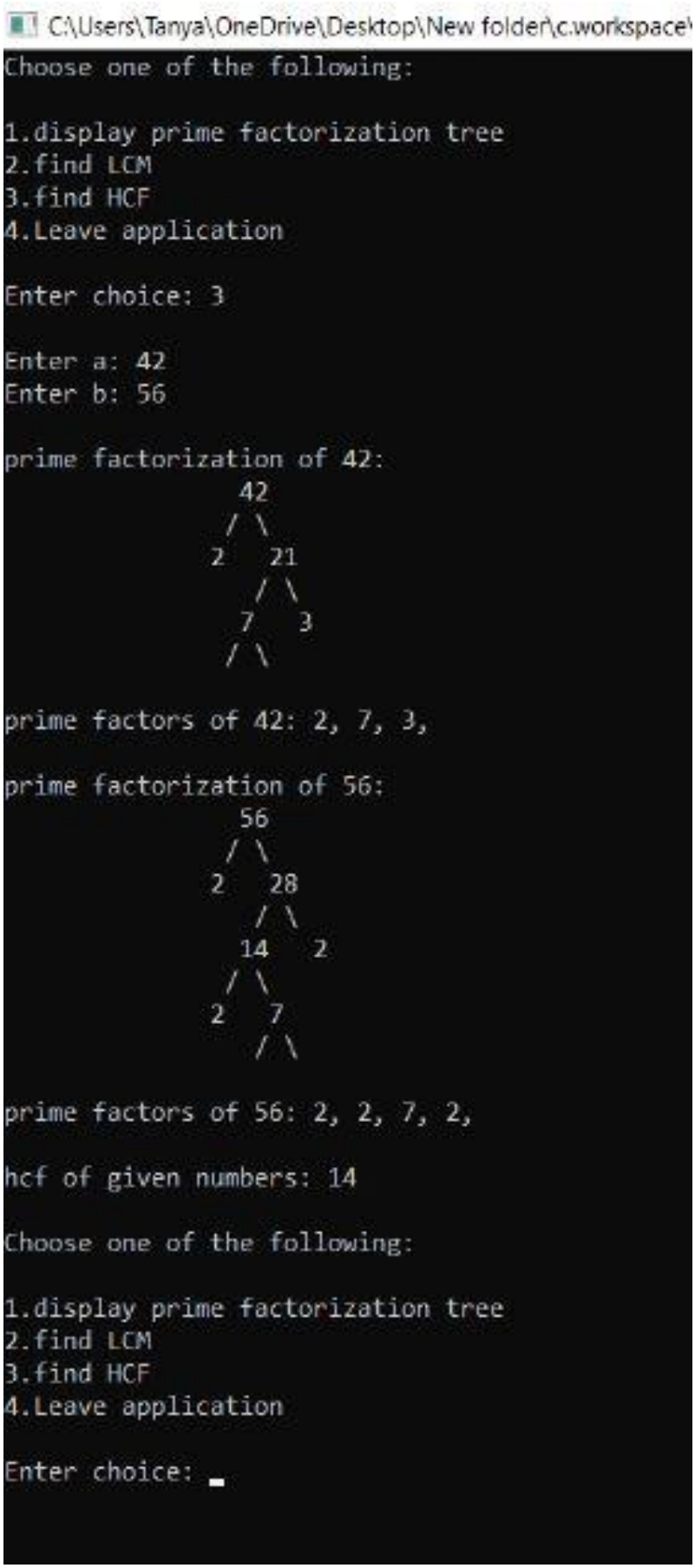
```
scanf("%d",&choice);
if(choice==1)
        {
   printf("\nEnter a natural number: ");
   scanf("%d",&a);
   printf("\nprime factorization tree of %d:\n\n",a);
            create(a);
   display();
   printf("\nprime factors of %d: ",a);
   ap=leafnode(root);
   for(i=0;i<1000;i++)
   {
      if(arr[i]!=0)
         arr[i]=0;
      else
         break;
   }
   printf("\n\n");
}
else if(choice==2)
        {
   printf("\nEnter a: ");
   scanf("%d",&a);
   printf("Enter b: ");
   scanf("%d",&b);
   printf("\nprime factorization of %d:\n",a);
   create(a);
   display();
   printf("\nprime factors of %d: ",a);
   ap=leafnode(root);
```

```
    for(i=0;i<1000;i++)
    {
       if(arr[i]!=0)
       {
          arp[i]=arr[i];
          arr[i]=0;
       }
       else
          break;
    }
    printf("\n\nprime factorization of %d:\n",b);
    create(b);
    display();
    printf("\nprime factors of %d: ",b);
    bp=leafnode(root);
    for(i=0;i<1000;i++)
    {
       if(arr[i]!=0)
       {
          brp[i]=arr[i];
          arr[i]=0;
       }
       else
          break;
    }
    printf("\n\nlcm of given numbers: ");
    findLCM(arp,brp);
    printf("\n\n");
  }
 else if(choice==3)
```

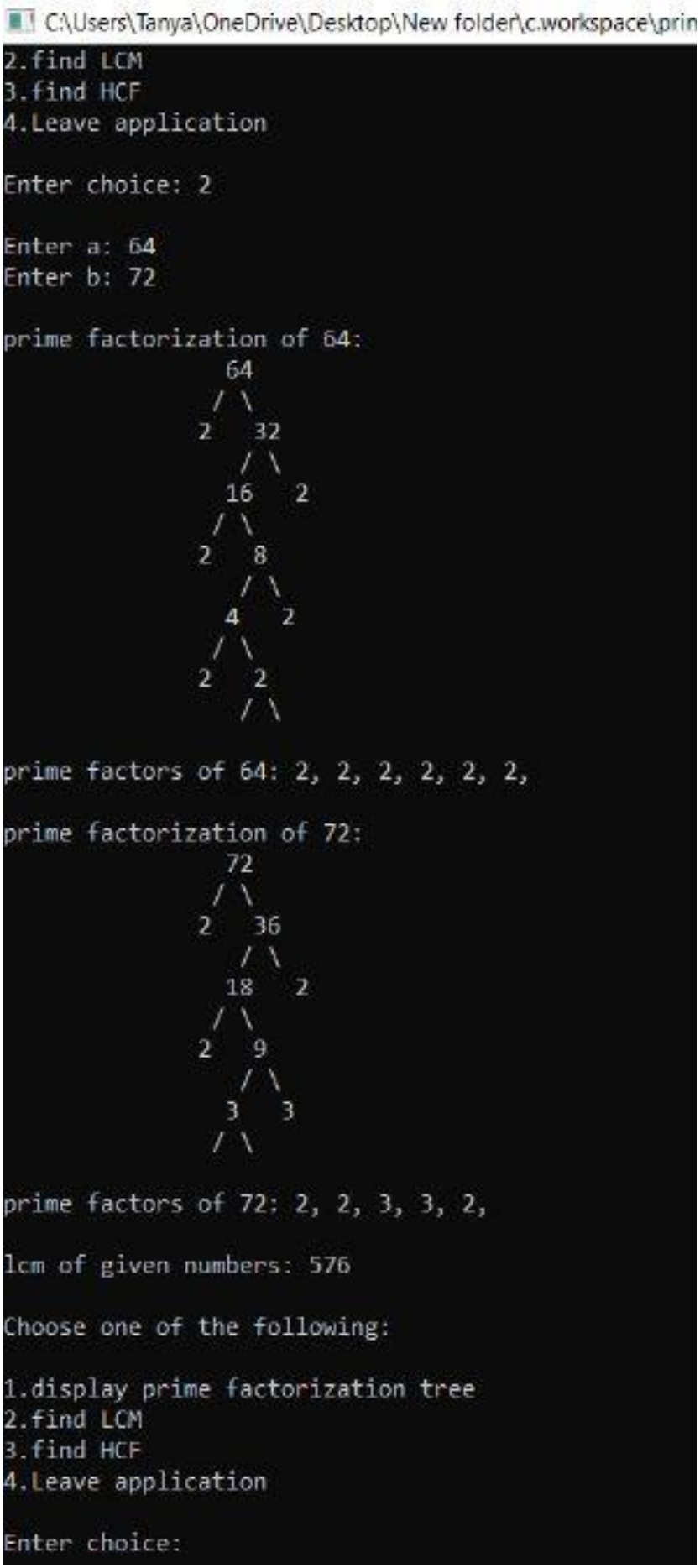
```
            {
printf("\nEnter a: ");
scanf("%d",&a);
printf("Enter b: ");
scanf("%d",&b);
printf("\nprime factorization of %d:\n",a);
create(a);
display();
printf("\nprime factors of %d: ",a);
ap=leafnode(root);
for(i=0;i<1000;i++)
{
   if(arr[i]!=0)
   {
      arp[i]=arr[i];
      arr[i]=0;
   }
   else
      break;
}
printf("\n\nprime factorization of %d:\n",b);
create(b);
display();
printf("\nprime factors of %d: ",b);
bp=leafnode(root);
for(i=0;i<1000;i++)
{
   if(arr[i]!=0)
   {
      brp[i]=arr[i];
```

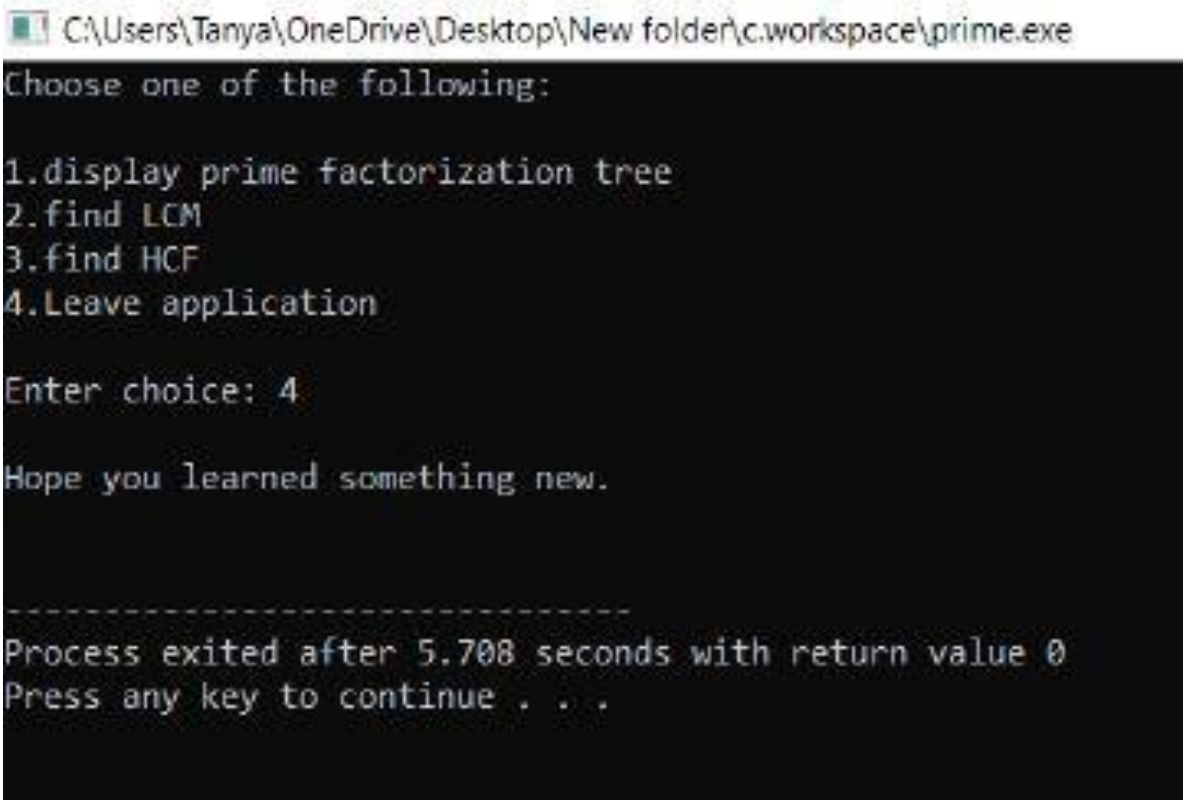
```
            arr[i]=0;
        }
        else
            break;
    }
    printf("\n\nhcf of given numbers: ");
    findHCF(arp,brp);
    printf("\n\n");
  }
  else if(choice==4)
            {

    printf("\nHope you learned something new.");
    f=0;
    printf("\n\n");
  }
  else
    printf("\ninvalid choice\n\n");
    }
}
```

Output:


C:\Users\Tanya\OneDrive\Desktop\New folder\c.workspace\
Choose one of the following:
1.display prime factorization tree
2.find LCM
3.find HCF
4.Leave application
Enter choice: 3
Enter a: 42
Enter b: 56
prime factorization of 42:
42
2 21
7 3
prime factors of 42: 2, 7, 3,
prime factorization of 56:
56
2 28
14 2
2 7
prime factors of 56: 2, 2, 7, 2,
hcf of given numbers: 14
Choose one of the following:
1.display prime factorization tree
2.find LCM
3.find HCF
4.Leave application
Enter choice:

```
C:\Users\Tanya\OneDrive\Desktop\New folder\c.work
Choose one of the following:

1.display prime factorization tree
2.find LCM
3.find HCF
4.Leave application

Enter choice: 1

Enter a natural number: 150

prime factorization tree of 150:

                150
               / \
              2   75
                 / \
                25   3
               / \
              5   5
                 / \

prime factors of 150: 2, 5, 5, 3,

Choose one of the following:

1.display prime factorization tree
2.find LCM
3.find HCF
4.Leave application

Enter choice:
```


C:\Users\Tanya\OneDrive\Desktop\New folder\c.workspace\prin
2.find LCM
3.find HCF
4.Leave application
Enter choice: 2
Enter a: 64
Enter b: 72
prime factorization of 64:
64
2 32
16 2
2 8
4 2
2 2
prime factors of 64: 2, 2, 2, 2, 2, 2,
prime factorization of 72:
72
2 36
18 2
2 9
3 3
prime factors of 72: 2, 2, 3, 3, 2,
lcm of given numbers: 576
Choose one of the following:
1.display prime factorization tree
2.find LCM
3.find HCF
4.Leave application
Enter choice:


C:\Users\Tanya\OneDrive\Desktop\New folder\c.workspace\prime.exe
Choose one of the following:
1.display prime factorization tree
2.find LCM
3.find HCF
4.Leave application
Enter choice: 4
Hope you learned something new.

Process exited after 5.708 seconds with return value 0
Press any key to continue . . .

Project 7: Graph

AIRLINE MAPPING AND BEST ROUTE RECOMMENDATION SYSTEM

Code:

```
//task - create a program that uses graphs with real-life applications
//TRANSPORTATION MAPPING AND BEST ROUTE RECOMMENDATION SYSTEM

#include<stdio.h>
#include<stdlib.h>
#include<string.h>

//REHA

//structure for edge nodes (each edge represents a flight)
struct edge
{
   float length;
   int time;
   int price;
   float av;
   int dest_code;
   int source_code;
   struct edge* link;
};

//method declaration
int add_loc(char[100][60], int);
int del_loc(char[100][60],int,struct edge*[]);
void upd_edge(char[100][60],int,struct edge*[]);
```

```
void add_edge(char[100][60],int,struct edge*[]);

void del_edge(char[100][60],int,struct edge*[]);

void readfile(char[100][60],int,struct edge*[]);

void writefile(char[100][60],int,struct edge*[]);

void allavail(int,int,char[100][60],int,struct edge*[]);

void shortestd(int,int,char[100][60],int,struct edge*[]);

void shortestt(int,int,char[100][60],int,struct edge*[]);

void shortestp(int,int,char[100][60],int,struct edge*[]);

void shortesta(int,int,char[100][60],int,struct edge*[]);



void main()

{

    int user,w=1,v,choice,source,destination,p_no=0,i;  //p_no is total places/locations

    char
p_code[100][60]={"","","","","","","","","","","","","","","","","","","","","","","","","","","",""
,"","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","",""};

    struct edge* arr[100];

    readfile(p_code,p_no,arr);

    printf("Welcome to Airline Flight Database!!\n");

    while(w)

    {

        printf("\nInformation you require before proceeding:\nLocations and associated
codes:\n");

        for(i=0;i<100;i++)

        {

            if(strcmp(p_code[i],""))

                printf("%d - %s\n",i+1,p_code[i]);

        }

        printf("\nChoose:\n1.Authorized personnel (to edit the information\n2.Customer (to see
available flight details)\n3.Leave Application\nEnter your choice: ");

        scanf("%d",&user);
```

```
if(user==1)
{
   v=1;
   while(v)
   {
      printf("\nDo you want to:\n1.Add a location\n2.Delete a location\n3.Add a flight\n4.Delete a flight\n5.Edit details of a flight\n6.Go back to home page\nEnter choice: ");
      scanf("%d",&choice);
      if(choice==1)
      {
         p_no=add_loc(p_code,p_no);
      }
      else if(choice==2)
      {
         p_no=del_loc(p_code,p_no,arr);
      }
      else if(choice==3)
      {
         add_edge(p_code,p_no,arr);
      }
      else if(choice==4)
      {
         del_edge(p_code,p_no,arr);
      }
      else if(choice==5)
      {
         upd_edge(p_code,p_no,arr);
      }
      else if(choice==6)
      {
         v=0;
```

```
            }
            else
                printf("\nChoice not available! Please try again with another choice.\n");
        }
    }
    else if(user==2)
    {
        v=1;
        while(v)
        {
            printf("\nEnter source place code: ");
            scanf("%d",&source);
            printf("\nEnter destination place code: ");
            scanf("%d",&destination);
            if((source>=1)&&(source<=100)&&(destination>=1)&&(destination<=100))
            {
                if((strcmp(p_code[source-1],""))&&(strcmp(p_code[destination-1],"")))
                {
                    printf("\nChoose the option you want to see:\n1.All flights
available\n2.Shortest route available\n3.Cheapest route available\n4.Fastest route
available\n5.Aggregated best rout available\n6.Go back to home page\nEnter choice: ");
                    scanf("%d",&choice);
                    if(choice==1)
                    {
                        allavail(source,destination,p_code,p_no,arr);
                    }
                    else if(choice==2)
                    {
                        shortestd(source,destination,p_code,p_no,arr);
                    }
                    else if(choice==3)
```

```
            {
                shortestp(source,destination,p_code,p_no,arr);
            }
            else if(choice==4)
            {
                shortestt(source,destination,p_code,p_no,arr);
            }
            else if(choice==5)
            {
                shortesta(source,destination,p_code,p_no,arr);
            }
            else if(choice==6)
            {
                v=0;
            }
            else
                printf("\nChoice not available! Please try again with another choice.\n");
        }
        else
        {
            printf("\nProvided code(s) do(es) not exist!\n");
            v=0;
        }
    }
    else
    {
        printf("\nProvided code(s) do(es) not exist!\n");
        v=0;
    }
}
```

```
        }
        else if(user=3)
        {
            w=0;
        }
        else
        {
            printf("\nChoice not available! Please try again with another choice.\n");
        }
    }
    writefile(p_code,p_no,arr);
}

//function to add a place/location
int add_loc(char p_code[100][60],int p_no)
{
    if(p_no==100)
    {
        printf("\nThe database cannot contain more than 100 locations at a time. Sorry.\n");
        return(p_no);
    }
    char name[60];
    printf("\nEnter name of new location(max. 60 characters): ");
    scanf("%s",name);
    int i=0;
    while(strcmp(p_code[i],""))
        i++;
    strcpy(p_code[i],name);
    printf("\n%s has been added successfully and its location code will be: %d\n",name,i+1);
    return(p_no+1);
```

```
}

//function to delete a location and all its associates flights
int del_loc(char p_code[100][60],int p_no,struct edge* arr[])
{
   if(p_no==0)
   {
      printf("\nSystem contains 0 locations so we cannot delete anything!\n");
      return(p_no);
   }
   int code;
   printf("\nenter location code of location to be deleted: ");
   scanf("%d",&code);
   if((code>=1)&&(code<=100))
   {
      if(strcmp(p_code[code-1],""))
      {
         arr[code-1]=NULL;
         int i;
         struct edge* p;
         for(i=0;i<100;i++)
         {
            p=arr[i];
            if(p==NULL)
               continue;
            if(p->dest_code==code)
               arr[i]=p->link;
            while(p->link!=NULL)
            {
               if(p->link->dest_code==code)
```

```
                p->link=p->link->link;
            p=p->link;
        }
    }
    printf("\nThe location %s and all the flights coming and going from it have been
successfully deleted from the database.\n",p_code[code-1]);
    strcpy(p_code[code-1],"");
    return(p_no-1);
  }
  else
  {
    printf("\nNo location is associated with this code, so cannot delete!\n");
    return(p_no);
  }
 }
 else
 {
  printf("\nThis location code does not exist!\n");
  return(p_no);
 }
}

//function to change details of particular flight
void upd_edge(char p_code[100][60],int p_no,struct edge* arr[100])
{
 int source,dest,ch;
 printf("\nEnter location code of source of flight: ");
 scanf("%d",&source);
 printf("\nEnter location code of destination of flight: ");
 scanf("%d",&dest);
 if((source>=1)&&(source<=100)&&(dest>=1)&&(dest<=100))
```

```
  {
    if((strcmp(p_code[source-1],""))&&(strcmp(p_code[dest-1],"")))
    {
      struct edge* p;
      p=arr[source-1];
      while(p!=NULL)
      {
        if(p->dest_code==dest)
          break;
        p=p->link;
      }
      if(p!=NULL)
      {
        printf("\nChoose the detail you want to change:\n1.Distance
covered\n2.Price\n3.Time taken\n4.Go back to admin home page\nEnter choice: ");
        scanf("%d",&ch);
        if(ch==1)
        {
          float d;
          printf("\nenter new value (in kilometres): ");
          scanf("%f",&d);
          p->av+=0.2*(d-(p->length));
          p->length=d;
        }
        else if(ch==2)
        {
          int pr;
          printf("\nenter new price value (in ruppees): ");
          scanf("%d",&pr);
          p->av+=0.4*(pr-(p->price));
          p->price=pr;
```

```
            }
            else if(ch==3)
            {
               int t;
               printf("\nenter new time taken value (format: hhmm, example: 2017 is 20 hours
and 17 minutes): ");
               scanf("%d",&t);
               p->av+=0.4*(t-(p->time));
               p->time=t;
            }
            else if(ch==4){}
            else
               printf("\nChoice not available!.\n");
         }
         else
            printf("\nNo flight exists from %s(%d) to %s(%d).\n",p_code[source-
1],source,p_code[dest-1],dest);
      }
       else
         printf("\nProvided code(s) do(es) not exist!\n");
   }
    else
      printf("\nProvided code(s) do(es) not exist!\n");
}

void readfile(char p_code[100][60],int p_no,struct edge* arr[100])
{
   FILE* f;
   f=fopen("AirlineLocations.txt","r");
   int n,s,d,t,p,i=0;
   float l,a;
```

```
    char st[60];
    while(fscanf(f,"%s %d\n",st,&n)!=EOF)
    {
        strcpy(p_code[n-1],st);
        p_no+=1;
    }
    for(i=0;i<100;i++)
        arr[i]=NULL;
    fclose(f);
    f=fopen("FlightDetails.txt","r");
    while(fscanf(f,"%d %d %f %d %d %f\n",&s,&d,&l,&t,&p,&a)!=EOF)
    {
        struct edge* ed;
        ed=(struct edge*)malloc(sizeof(struct edge));
        ed->av=a;
        ed->dest_code=d;
        ed->length=l;
        ed->price=p;
        ed->source_code=s;
        ed->time=t;
        ed->link=arr[s-1];
        arr[s-1]=ed;
    }
    fclose(f);
}

void writefile(char p_code[100][60],int p_no,struct edge* arr[100])
{
    int i;
    FILE* f;
```

```
    f=fopen("AirlineLocations.txt","w");
    for(i=0;i<100;i++)
    {
        if(strcmp(p_code[i],""))
            fprintf(f,"%s %d\n",p_code[i],i+1);
    }
    fclose(f);
    f=fopen("FlightDetails.txt","w");
    struct edge* p;
    for(i=0;i<100;i++)
    {
        p=arr[i];
        while(p!=NULL)
        {
            fprintf(f,"%d %d %f %d %d %f\n",p->source_code,p->dest_code,p->length,p-
>time,p->price,p->av);
            p=p->link;
            printf("123");
        }
    }
    fclose(f);
}

void allavail(int source,int dest,char p_code[100][60],int p_no,struct edge* arr[100])
{
    struct edge* p;
    int i=1;
    p=arr[source-1];
    while(p!=NULL)
    {
        if(p->dest_code==dest)
```

```
        {
            printf("\nFlight %d:\nPath length: %f\nCost: %d\nTime taken: %d\n",i,p->length,p->price,p->time);
            i++;
        }
        p=p->link;
    }
}

//TANYA

void add_edge(char p_code[100][60],int p_no,struct edge* arr[100])
{
        int p,t,d,s;
        float dist,avg;
        struct edge*ptr;
        printf("\nEnter location code of source: ");
        scanf("%d" ,&s);
        printf("Enter location code of destination: ");
        scanf("%d",&d);
        if((s>=1)&&(s<=100)&&(d>=1)&&(d<=100))
    {
        if((strcmp(p_code[s-1],""))&&(strcmp(p_code[d-1],"")))
                    {
        printf("Enter distance covered by flight: ");
        scanf("%f",&dist);
        printf("Enter price: ");
        scanf("%d",&p);
        printf("Enter time taken(format: hhmm, example: 2017 is 20 hours and 17 minutes): ");
        scanf("%d",&t);
```

```
        avg=(t*0.4+d*0.2+p*0.4);
        ptr=(struct edge*)malloc(sizeof(struct edge));
        ptr->source_code=s;
        ptr->length=dist;
                ptr->time=t;
                        ptr->price=p;
                ptr->av=avg;
                ptr->dest_code=d;
                ptr->source_code=s;
        ptr->link=arr[s-1];
        arr[s-1]=ptr;
        printf("\nSuccessfully added the flight!\n");
                }
        }
}

void del_edge(char p_code[100][60],int p_no,struct edge* arr[100])
{
        int p,t,d,s;
        float dist,avg;
        printf("\nEnter location code of source: ");
        scanf("%d",&s);
        printf("Enter location code of destination: ");
        scanf("%d",&d);
        printf("Enter distance covered by flight: ");
        scanf("%f",&dist);
        printf("Enter price: ");
   scanf("%d",&p);
   printf("Enter time taken(format: hhmm, example: 2017 is 20 hours and 17 minutes): ");
   scanf("%d",&t);
```

```
        if((s>=1)&&(s<=100)&&(d>=1)&&(d<=100))
   {
        if((strcmp(p_code[s-1],""))&&(strcmp(p_code[d-1],"")))
                {
                struct edge* ed;
        ed=arr[s-1];
        if(ed==NULL)
                printf("\nNo flight starts at given location!\n");
                        else
                        {
                           int x=0;
                while(ed->link!=NULL)
                {
                        if((ed->link->dest_code==d)&&(ed->link->price==p)&&(ed->link->length==dist)&&(ed->link->time==t))
                                        {
                                ed->link==ed->link->link;
                                printf("\nIt has been deleted successfully.\n");
                                x=1;
                                break;
                                        }
                                        ed=ed->link;
                                }
                                if(x==0)
                                        printf("Such a flight doesn't exist!\n");
        }
                }
                else
        printf("\nProvided code(s) do(es) not exist!\n");
        }
        else
```

```
        printf("\nProvided code(s) do(es) not exist!\n");
}


//KRUPA AND MISARI

//void readfile(char p_code[100][60],int p_no,struct edge* arr[100]){}

//void writefile(char p_code[100][60],int p_no,struct edge* arr[100]){}

//VRUNDA (WITH REHA)

void shortestd(int s,int d,char p_code[100][60],int p_no,struct edge* arr[100])
{
   int i,j,k,s1,d1;
   struct edge* p;
   float mat[p_no][p_no];
   for(i=0;i<p_no;i++)
   {
      for(j=0;j<p_no;j++)
         mat[i][j]=0;
   }
   j=0;
   for(i=0;i<100;i++)
   {
      if(strcmp(p_code[i],""))
      {
         p=arr[i];
         while(p!=NULL)
         {
```

```
            if((mat[j][p->dest_code-1]==0)||(mat[j][p->dest_code-1]>p->length))
               mat[j][p->dest_code-1]=p->length;
            p=p->link;
         }
         if(i==s-1)
            s1=j;
         if(i==d-1)
            d1=j;
         j++;
      }
   }
   float mat1[p_no][p_no];
   for(i=0;i<p_no;i++)
   {
      for(j=0;j<p_no;j++)
      {
         for(k=0;k<p_no;k++)
         {
            mat1[j][k]=mat[j][k];
            if((mat[j][i]!=0)&&(mat[i][k]!=0))
            {
               if((mat[j][k]==0)||(mat[j][k]>(mat[j][i]+mat[i][k])))
                  mat1[j][k]=mat[j][i]+mat[i][k];
            }
         }
      }
      for(j=0;i<p_no;i++)
      {
         for(k=0;j<100;j++)
            mat[j][k]=mat1[j][k];
```

```
        }
    }
    printf("%f",mat[s1][d1]);
    if(mat[s1][d1]==0)
        printf("\nNo path available from %s(%d) to %s(%d).\n",p_code[s-1],s,p_code[d-1],d);
    else
        printf("\nShortest distance path from %s(%d) to %s(%d): %f\n",p_code[s-1],s,p_code[d-1],d,mat[s1][d1]);
}

void shortestt(int s,int d,char p_code[100][60],int p_no,struct edge* arr[100])
{
    int i,j,k,s1,d1;
    struct edge* p;
    int mat[p_no][p_no];
    for(i=0;i<p_no;i++)
    {
        for(j=0;j<p_no;j++)
            mat[i][j]=0;
    }
    j=0;
    for(i=0;i<100;i++)
    {
        if(strcmp(p_code[i],""))
        {
            p=arr[i];
            while(p!=NULL)
            {
                if((mat[j][p->dest_code-1]==0)||(mat[j][p->dest_code-1]>p->time))
                    mat[j][p->dest_code-1]=p->time;
                p=p->link;
```

```
        }
        if(i==s-1)
            s1=j;
        if(i==d-1)
            d1=j;
        j++;
    }
}
int mat1[p_no][p_no];
for(i=0;i<p_no;i++)
{
    for(j=0;j<p_no;j++)
    {
        for(k=0;k<p_no;k++)
        {
            mat1[j][k]=mat[j][k];
            if((mat[j][i]!=0)&&(mat[i][k]!=0))
            {
                if((mat[j][k]==0)||(mat[j][k]>(mat[j][i]+mat[i][k])))
                    mat1[j][k]=mat[j][i]+mat[i][k];
            }
        }
    }
    for(j=0;i<p_no;i++)
    {
        for(k=0;j<100;j++)
            mat[j][k]=mat1[j][k];
    }
}
if(mat[s1][d1]==0)
```

```
    printf("\nNo path available from %s(%d) to %s(%d).\n",p_code[s-1],s,p_code[d-1],d);
  else
    printf("\nFastest path time from %s(%d) to %s(%d): %d\n",p_code[s-1],s,p_code[d-1],d,mat[s1][d1]);
}

void shortestp(int s,int d,char p_code[100][60],int p_no,struct edge* arr[100])
{
  int i,j,k,s1,d1;
  struct edge* p;
  int mat[p_no][p_no];
  for(i=0;i<p_no;i++)
  {
    for(j=0;j<p_no;j++)
      mat[i][j]=0;
  }
  j=0;
  for(i=0;i<100;i++)
  {
    if(strcmp(p_code[i],""))
    {
      p=arr[i];
      while(p!=NULL)
      {
        if((mat[j][p->dest_code-1]==0)||(mat[j][p->dest_code-1]>p->price))
          mat[j][p->dest_code-1]=p->price;
        p=p->link;
      }
      if(i==s-1)
        s1=j;
      if(i==d-1)
```

```
            d1=j;
        j++;
    }
  }
  int mat1[p_no][p_no];
  for(i=0;i<p_no;i++)
  {
    for(j=0;j<p_no;j++)
    {
      for(k=0;k<p_no;k++)
      {
        mat1[j][k]=mat[j][k];
        if((mat[j][i]!=0)&&(mat[i][k]!=0))
        {
          if((mat[j][k]==0)||(mat[j][k]>(mat[j][i]+mat[i][k])))
            mat1[j][k]=mat[j][i]+mat[i][k];
        }
      }
    }
    for(j=0;i<p_no;i++)
    {
      for(k=0;j<100;j++)
        mat[j][k]=mat1[j][k];
    }
  }
  if(mat[s1][d1]==0)
    printf("\nNo path available from %s(%d) to %s(%d).\n",p_code[s-1],s,p_code[d-1],d);
  else
    printf("\nCheapest path price from %s(%d) to %s(%d): %d\n",p_code[s-1],s,p_code[d-
1],d,mat[s1][d1]);
}
```

```
void shortesta(int s,int d,char p_code[100][60],int p_no,struct edge* arr[100])
{
   int i,j,k,s1,d1;
   struct edge* p;
   float mat[p_no][p_no];
   for(i=0;i<p_no;i++)
   {
      for(j=0;j<p_no;j++)
         mat[i][j]=0;
   }
   j=0;
   for(i=0;i<100;i++)
   {
      if(strcmp(p_code[i],""))
      {
         p=arr[i];
         while(p!=NULL)
         {
            if((mat[j][p->dest_code-1]==0)||(mat[j][p->dest_code-1]>p->av))
               mat[j][p->dest_code-1]=p->av;
            p=p->link;
         }
         if(i==s-1)
            s1=j;
         if(i==d-1)
            d1=j;
         j++;
      }
   }
```

```
    float mat1[p_no][p_no];
    for(i=0;i<p_no;i++)
    {
        for(j=0;j<p_no;j++)
        {
            for(k=0;k<p_no;k++)
            {
                mat1[j][k]=mat[j][k];
                if((mat[j][i]!=0)&&(mat[i][k]!=0))
                {
                    if((mat[j][k]==0)||(mat[j][k]>(mat[j][i]+mat[i][k])))
                        mat1[j][k]=mat[j][i]+mat[i][k];
                }
            }
        }
        for(j=0;i<p_no;i++)
        {
            for(k=0;j<100;j++)
                mat[j][k]=mat1[j][k];
        }
    }
    printf("%f",mat[s1][d1]);
    if(mat[s1][d1]!=0)
        printf("\nNo path available from %s(%d) to %s(%d).\n",p_code[s-1],s,p_code[d-1],d);
    else
        printf("\nBest path (acc. to aggregated formula) value from %s(%d) to %s(%d):
%f\n",p_code[s-1],s,p_code[d-1],d,mat[s1][d1]);
}
```

Output:

AirlineLocations.txt - Notepad

File Edit Format View Help

```
Surat 1
Mumbai 2
Chennai 3
Kolkata 4
```

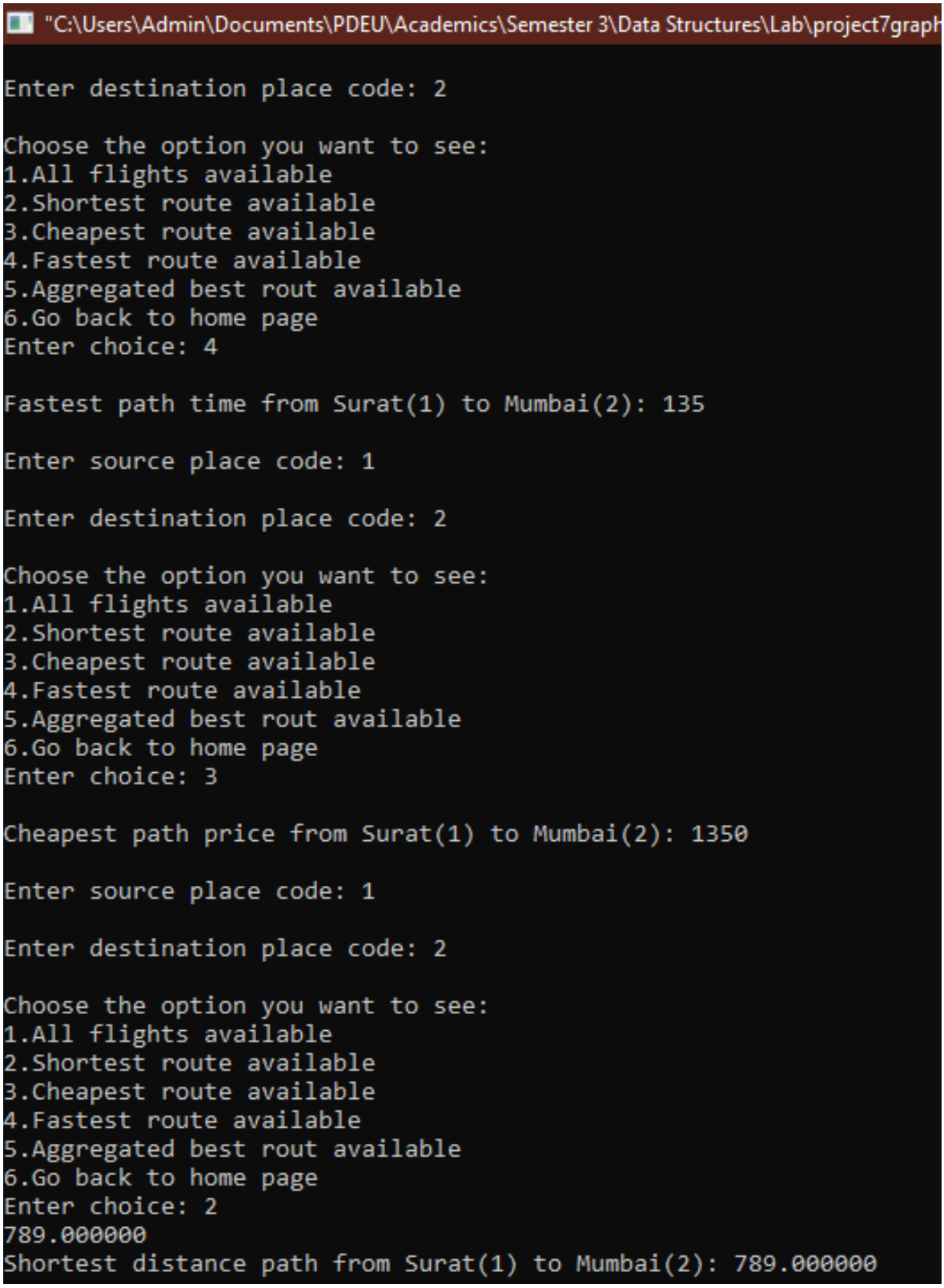
FlightDetails.txt - Notepad

File Edit Format View Help

```
1 2 789.000000 135 1750 754.400024
1 2 900.000000 213 1350 625.599976
```

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\

```
Welcome to Airline Flight Database!!

Information you require before proceeding:
Locations and associated codes:
1 - Surat
2 - Mumbai
3 - Chennai
4 - Kolkata

Choose:
1.Authorized personnel (to edit the information
2.Customer (to see available flight details)
3.Leave Application
Enter your choice: 2

Enter source place code: 1

Enter destination place code: 2
```

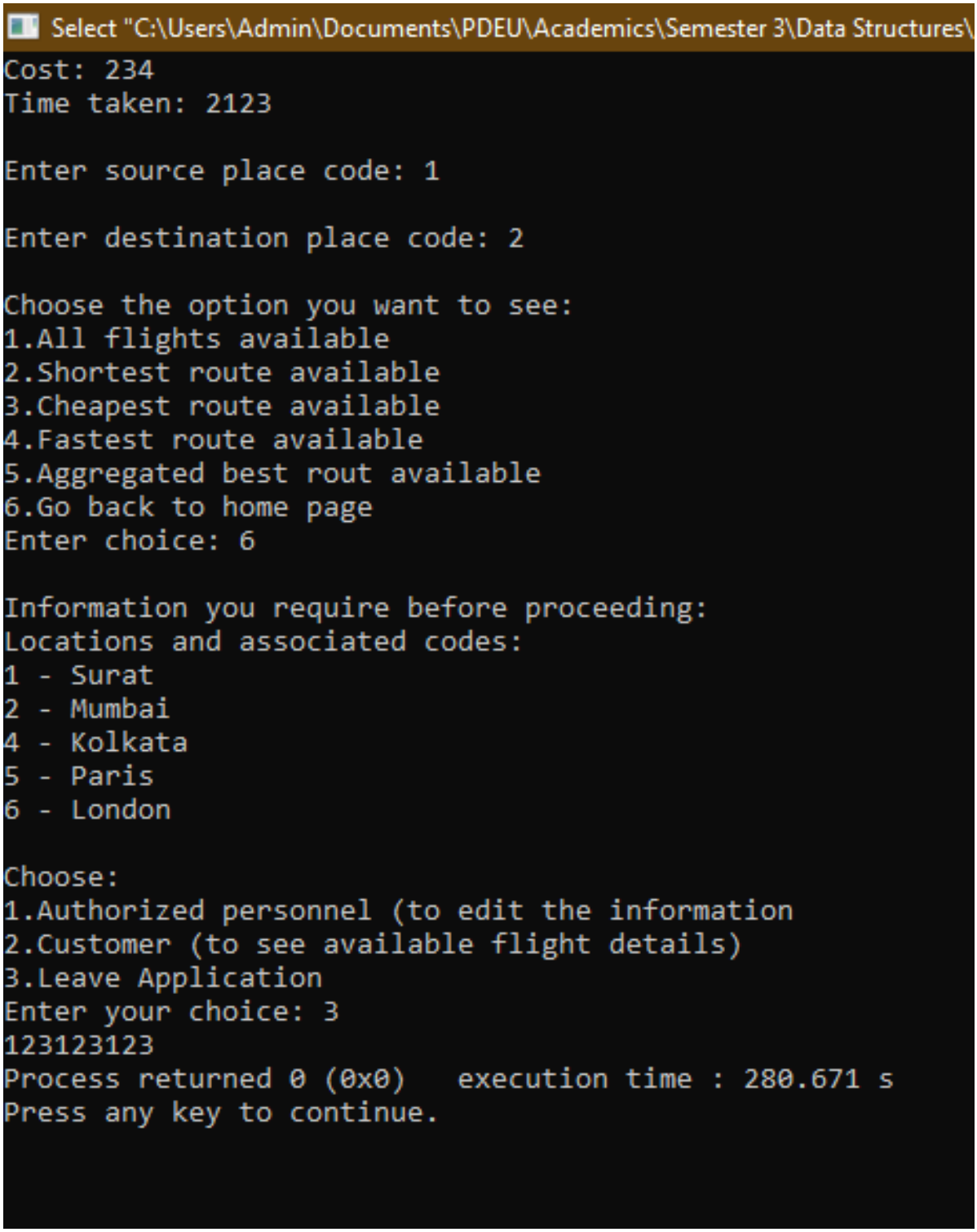

"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project7graph
Enter destination place code: 2
Choose the option you want to see:
1.All flights available
2.Shortest route available
3.Cheapest route available
4.Fastest route available
5.Aggregated best rout available
6.Go back to home page
Enter choice: 4
Fastest path time from Surat(1) to Mumbai(2): 135
Enter source place code: 1
Enter destination place code: 2
Choose the option you want to see:
1.All flights available
2.Shortest route available
3.Cheapest route available
4.Fastest route available
5.Aggregated best rout available
6.Go back to home page
Enter choice: 3
Cheapest path price from Surat(1) to Mumbai(2): 1350
Enter source place code: 1
Enter destination place code: 2
Choose the option you want to see:
1.All flights available
2.Shortest route available
3.Cheapest route available
4.Fastest route available
5.Aggregated best rout available
6.Go back to home page
Enter choice: 2
789.000000
Shortest distance path from Surat(1) to Mumbai(2): 789.000000

```
"C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project7graph.exe"

Enter source place code: 1

Enter destination place code: 2

Choose the option you want to see:
1.All flights available
2.Shortest route available
3.Cheapest route available
4.Fastest route available
5.Aggregated best rout available
6.Go back to home page
Enter choice: 6

Information you require before proceeding:
Locations and associated codes:
1 - Surat
2 - Mumbai
3 - Chennai
4 - Kolkata

Choose:
1.Authorized personnel (to edit the information
2.Customer (to see available flight details)
3.Leave Application
Enter your choice: 1

Do you want to:
1.Add a location
2.Delete a location
3.Add a flight
4.Delete a flight
5.Edit details of a flight
6.Go back to home page
Enter choice: 1

Enter name of new location(max. 60 characters): Paris

Paris has been added successfully and its location code will be: 5

Do you want to:
1.Add a location
```

```
Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project7graph.exe"

Do you want to:
1.Add a location
2.Delete a location
3.Add a flight
4.Delete a flight
5.Edit details of a flight
6.Go back to home page
Enter choice: 1

Enter name of new location(max. 60 characters): London

London has been added successfully and its location code will be: 6

Do you want to:
1.Add a location
2.Delete a location
3.Add a flight
4.Delete a flight
5.Edit details of a flight
6.Go back to home page
Enter choice: 2

enter location code of location to be deleted: 3

The location Chennai and all the flights coming and going from it have been successfully deleted from the database.

Do you want to:
1.Add a location
2.Delete a location
3.Add a flight
4.Delete a flight
5.Edit details of a flight
6.Go back to home page
Enter choice: 3

Enter location code of source: 5
Enter location code of destination: 6
Enter distance covered by flight: 1234
Enter price: 234
Enter time taken(format: hhmm, example: 2017 is 20 hours and 17 minutes): 2123

Successfully added the flight!
```

```
Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\Lab\project7graph.exe"
Enter time taken(format: hhmm, example: 2017 is 20 hours and 17 minutes): 2123

Successfully added the flight!

Do you want to:
1.Add a location
2.Delete a location
3.Add a flight
4.Delete a flight
5.Edit details of a flight
6.Go back to home page
Enter choice: 6

Information you require before proceeding:
Locations and associated codes:
1 - Surat
2 - Mumbai
4 - Kolkata
5 - Paris
6 - London

Choose:
1.Authorized personnel (to edit the information
2.Customer (to see available flight details)
3.Leave Application
Enter your choice: 2
```

```
Select "C:\Users\Admin\Documents\PDEU\Academics\Semester
2.Customer (to see available flight details)
3.Leave Application
Enter your choice: 2

Enter source place code: 5

Enter destination place code: 6

Choose the option you want to see:
1.All flights available
2.Shortest route available
3.Cheapest route available
4.Fastest route available
5.Aggregated best rout available
6.Go back to home page
Enter choice: 1

Flight 1:
Path length: 1234.000000
Cost: 234
Time taken: 2123

Enter source place code: 1

Enter destination place code: 2

Choose the option you want to see:
1.All flights available
2.Shortest route available
3.Cheapest route available
4.Fastest route available
5.Aggregated best rout available
6.Go back to home page
Enter choice: 6

Information you require before proceeding:
Locations and associated codes:
1 - Surat
2 - Mumbai
4 - Kolkata
5 - Paris
6 - London

Choose:
```


Select "C:\Users\Admin\Documents\PDEU\Academics\Semester 3\Data Structures\
Cost: 234
Time taken: 2123
Enter source place code: 1
Enter destination place code: 2
Choose the option you want to see:
1.All flights available
2.Shortest route available
3.Cheapest route available
4.Fastest route available
5.Aggregated best rout available
6.Go back to home page
Enter choice: 6
Information you require before proceeding:
Locations and associated codes:
1 - Surat
2 - Mumbai
4 - Kolkata
5 - Paris
6 - London
Choose:
1.Authorized personnel (to edit the information
2.Customer (to see available flight details)
3.Leave Application
Enter your choice: 3
123123123
Process returned 0 (0x0) execution time : 280.671 s
Press any key to continue.

AirlineLocations.txt - Notepad

File Edit Format View Help

```
Surat 1
Mumbai 2
Kolkata 4
Paris 5
London 6
```

FlightDetails.txt - Notepad

File Edit Format View Help

```
1 2 900.000000 213 1350 625.599976
1 2 789.000000 135 1750 754.400024
5 6 1234.000000 2123 234 944.000000
```