**Department :** Computer Engineering
**Class :** SE
**Subject :** Data Structure Lab
**Name:** Kartiki Uday Khare.
**Roll no:** 21494
**Batch:** H4

# Assignment No : 08

- **Problem Statement:**

Write C++ program for storing appointment schedule for day. Appointments are booked randomly using linked list. Set start and end time and min and max duration for visit slot. Write functions for- a) Display free slots b) Book appointment c) Cancel appointment ( check validity, time bounds, availability) d) Sort list based on time e) Sort list based on time using pointer manipulation

- **Code:**

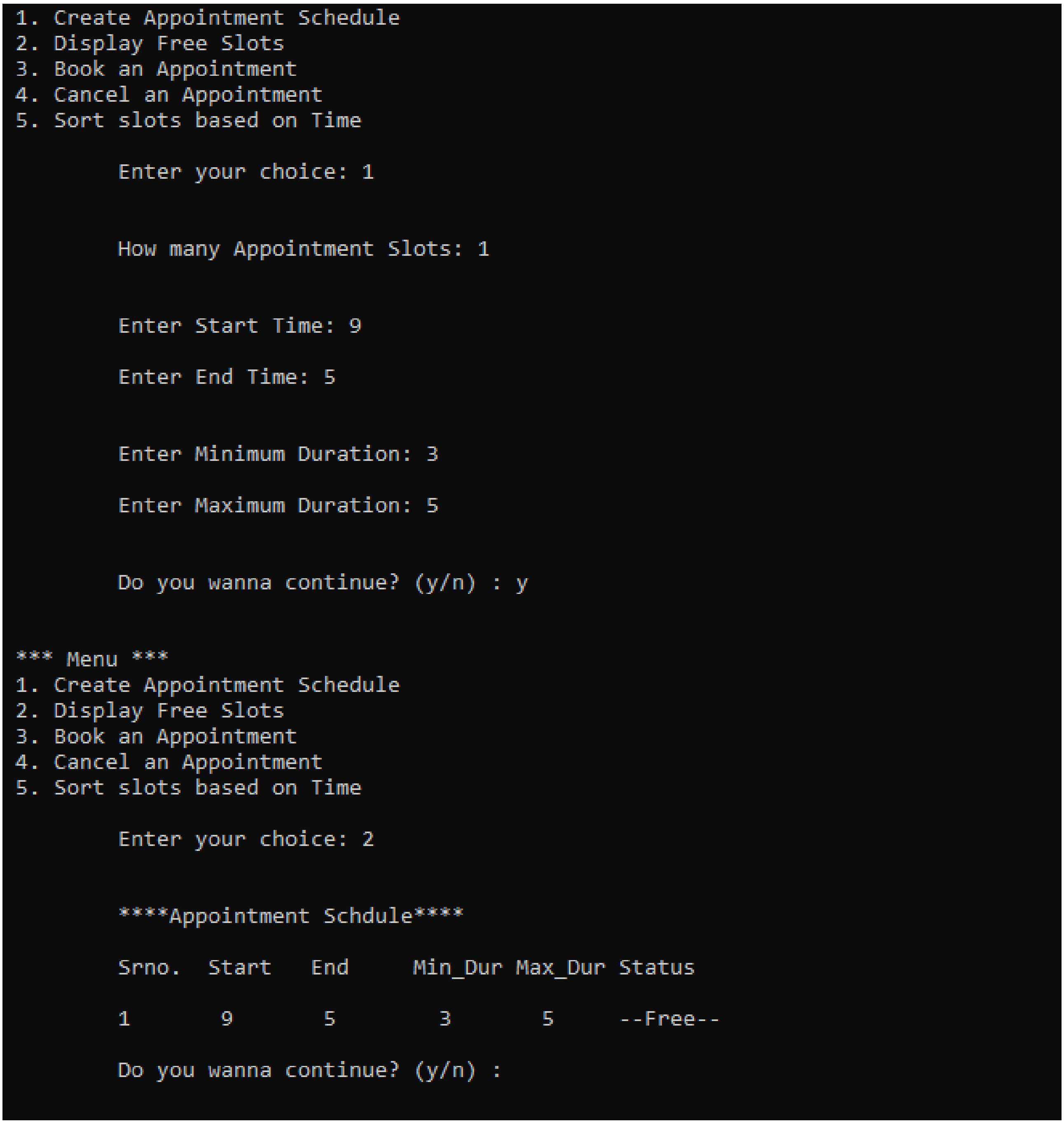
```
#include<iostream>
using namespace std;

int size; // No of Nodes or Appointments

struct SLL_Node// Node Structure of each Appointment
{
   int start;
   int end;
   int min;
   int max;
   int flag;
   struct SLL_Node *next;
}*head;


class App_Shedule
{
  public:
             void create_Shed();

             void display_Shed();

             void book_App();

             void cancel_App();

             void sort_App();

}A1;


int main()
{
   int ch;
   char ans;

   do
   {
     cout<<"\n\n *** Menu ***";
```

```
    cout<<"\n 1. Create Appointment Schedule";
    cout<<"\n 2. Display Free Slots";
    cout<<"\n 3. Book an Appointment";
    cout<<"\n 4. Cancel an Appointment";
    cout<<"\n 5. Sort slots based on Time";


    cout<<"\n\n\t Enter your choice: ";
    cin>>ch;

    switch(ch)
    {
      case 1: A1.create_Shed();
          break;

      case 2: A1.display_Shed();
          break;

      case 3: A1.book_App();
          break;

      case 4: A1.cancel_App();
          break;

      case 5: A1.sort_App();
          break;

      default: cout<<"\n\t Wrong choice!!!";

    }

    cout<<"\n\n\t Do you wanna continue? (y/n) : ";
    cin>>ans;
  }while(ans == 'y');

}


void App_Shedule :: create_Shed()          //Function Definition to create Appointment Schedule
{
   int i;
   struct SLL_Node *temp, *last;

   head = NULL;

   cout<<"\n\n\t How many Appointment Slots: ";
   cin>>size;

   for(i=0; i<size; i++)
   {
     temp = new(struct SLL_Node);        // Step 1: Dynamic Memory Allocation

     cout<<"\n\n\t Enter Start Time: ";   // Step 2: Assign Data & Address
     cin>>temp->start;
     cout<<"\n\t Enter End Time: ";
     cin>>temp->end;
     cout<<"\n\n\t Enter Minimum Duration: ";
     cin>>temp->min;
     cout<<"\n\t Enter Maximum Duration: ";
     cin>>temp->max;
     temp->flag = 0;
     temp->next = NULL;
     if(head == NULL)
     {
       head = temp;
       last = head;
     }
     else
     {
       last->next = temp;
       last = last->next;
     }

   }
}


void App_Shedule :: display_Shed()        //Function Definition to Display Appointment Schedule
{
  int cnt = 1;
  struct SLL_Node *temp;

  cout<<"\n\n\t ****Appointment Schdule****";
  cout<<"\n\n\t Srno.\tStart\tEnd\tMin_Dur\tMax_Dur\tStatus";

  temp = head;
  while(temp != NULL)
  {
    cout<<"\n\n\t "<<cnt;
    cout<<"\t "<<temp->start;
    cout<<"\t "<<temp->end;
    cout<<"\t  "<<temp->min;
    cout<<"\t  "<<temp->max;

    if(temp->flag)
      cout<<"\t-Booked-";
    else
      cout<<"\t--Free--";

    temp = temp->next;
    cnt++;
  }
}


void App_Shedule :: book_App()           //Function Definition to Book Appointment
{
   int start;
   struct SLL_Node *temp;

   cout<<"\n\n\t Please enter Appointment time: ";
   cin>>start;

  temp = head;

  while(temp != NULL)
  {
    if(start == temp->start)
    {
      if(temp->flag == 0)
      {
        cout<<"\n\n\t Appointment Slot is Booked!!!";
        temp->flag = 1;
      }
      else
```

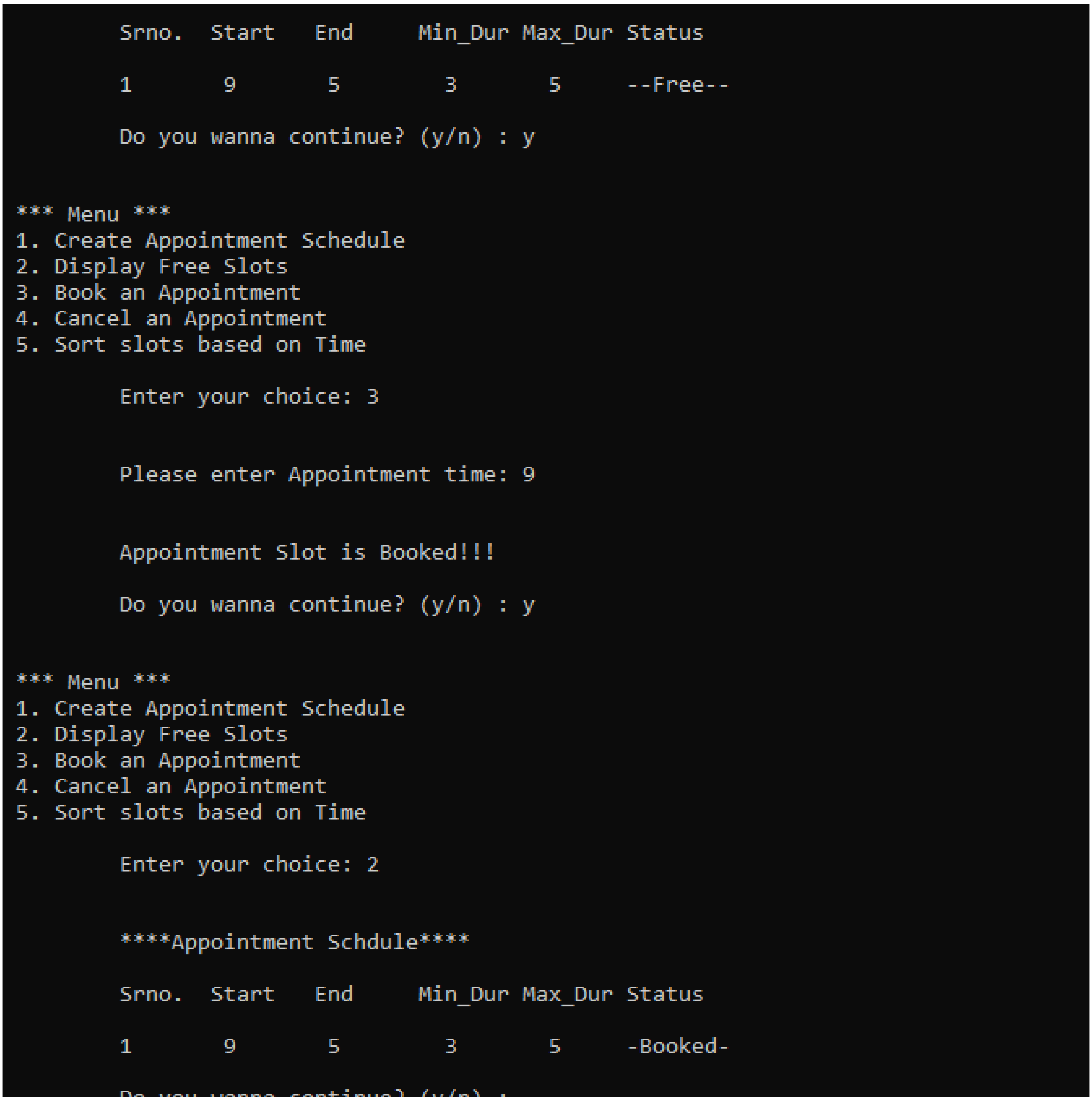
```
        cout<<"\n\n\t Appointment Slot is not Available!!!";
    }

    temp = temp->next;
  }

}



void App_Shedule :: cancel_App()        //Function Defination to Cancel Appointment
{
   int start;
   struct SLL_Node *temp;

   cout<<"\n\n\t Please enter Appointment time to Cancel: ";
   cin>>start;

  temp = head;

  while(temp != NULL)
  {
    if(start == temp->start)
    {
      if(temp->flag == 1)
      {
        cout<<"\n\n\t Your Appointment Slot is Canceled!!!";
        temp->flag = 0;
      }
      else
        cout<<"\n\n\t Your Appointment was not Booked!!!";
    }

    temp = temp->next;
  }
}


void App_Shedule :: sort_App()         //Function Definition to Sort Appointments
{
  int i,val;
  struct SLL_Node *temp;


  for(i=0; i < size-1; i++)
  {
     temp = head;
     while(temp->next != NULL)
     {
       if(temp->start > temp->next->start)
       {
          val = temp->start;
              temp->start = temp->next->start;
              temp->next->start = val;

          val = temp->end;
              temp->end = temp->next->end;
              temp->next->end = val;

          val = temp->min;
              temp->min = temp->next->min;
              temp->next->min = val;
```

```
            val = temp->max;
                temp->max = temp->next->max;
                temp->next->max = val;

          }
          temp = temp->next;
        }
    }

    cout<<"\n\n\t The Appointments got Sorted!!!";

}
```

- **Output:**



```
1. Create Appointment Schedule
2. Display Free Slots
3. Book an Appointment
4. Cancel an Appointment
5. Sort slots based on Time

        Enter your choice: 1


        How many Appointment Slots: 1


        Enter Start Time: 9

        Enter End Time: 5


        Enter Minimum Duration: 3

        Enter Maximum Duration: 5


        Do you wanna continue? (y/n) : y


*** Menu ***
1. Create Appointment Schedule
2. Display Free Slots
3. Book an Appointment
4. Cancel an Appointment
5. Sort slots based on Time

        Enter your choice: 2


        ****Appointment Schdule****

        Srno.  Start   End     Min_Dur Max_Dur Status

        1       9       5        3       5     --Free--

        Do you wanna continue? (y/n) :
```



```
        Srno.  Start   End     Min_Dur Max_Dur Status

        1       9       5        3       5     --Free--

        Do you wanna continue? (y/n) : y


*** Menu ***
1. Create Appointment Schedule
2. Display Free Slots
3. Book an Appointment
4. Cancel an Appointment
5. Sort slots based on Time

        Enter your choice: 3


        Please enter Appointment time: 9


        Appointment Slot is Booked!!!

        Do you wanna continue? (y/n) : y


*** Menu ***
1. Create Appointment Schedule
2. Display Free Slots
3. Book an Appointment
4. Cancel an Appointment
5. Sort slots based on Time

        Enter your choice: 2


        ****Appointment Schdule****

        Srno.  Start   End     Min_Dur Max_Dur Status

        1       9       5        3       5     -Booked-

        Do you wanna continue? (y/n) :
```