```
//DSL Lab 08 Appointment



#include<iostream>
using namespace std;

int numofapp;

struct node{

int minDuration;

int maxDuration;
int start;
int endt;
int duration;
int flag;
struct node *next;
struct node *prev;

};

class schedule{
public:
    node *head, *foot;
    schedule(){
        head = NULL;
        foot = NULL;
    }

    void scheduleEntry(){
    int i;
    struct node *temp, *last;

    head = NULL;

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

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

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

    }

    }

    void display(){
    int cnt = 1;
    struct node *temp;

   cout<<"\n\n <------ Appointment Schedule ------>";
   cout<<"\n\n
Sr.\tStart\tEnd\tMin_Duration\tMax_Duration\tDuration\tStatus";

   temp = head;
   while(temp != NULL)
   {
      cout<<"\n\n "<<cnt;
      cout<<"\t "<<temp->start;
      cout<<"\t "<<temp->endt;
      cout<<"\t "<<temp->minDuration;
      cout<<"\t\t "<<temp->maxDuration;
      cout<<"\t\t "<<temp->endt - temp->start;
      if(temp->flag)
         cout<<"\t\t--Booked--";
      else
         cout<<"\t\t--Free--";

      temp = temp->next;
      cnt++;
   }
    }
    void createAppoint(){
    int start;
    struct node *temp;

    cout<<"\n\n Please enter Appointment time: ";
    cin>>start;
    temp = head;

   while(temp != NULL)
   {
      if(start == temp->start)
      {
         if(temp->flag == 0)
         {
            cout<<"\n\n Congratulation, your appointment has been
scheduled!";
            temp->flag = 1;
         }
         else
            cout<<"\n\n Sorry, appointment slot is not available!";
      }

      temp = temp->next;
   }
    }

    void cancelAppoint(){
    int start;
    struct node *temp;

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

   temp = head;

   while(temp != NULL)
   {
      if(start == temp->start)
      {
         if(temp->flag == 1)
         {
            cout<<"\n\n Your appointment slot is cancelled!";
            temp->flag = 0;
         }
         else
            cout<<"\n\n Your appointment was not booked!";
      }

      temp = temp->next;
   }
    }

    void sortAppoint(){
    int i,j,val;
   struct node *temp;


   for(i=0; i < numofapp -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->endt;
              temp->endt = temp->next->endt;
              temp->next->endt = val;
              val = temp->minDuration;
              temp->minDuration= temp->next->minDuration;
              temp->next->minDuration = val;

              val = temp->maxDuration;
              temp->maxDuration = temp->next->maxDuration;
              temp->next->maxDuration = val;

          }
          temp = temp->next;
       }
   }

   cout<<"\n\nThe Appointments got Sorted!";

    }
};
int main(){
int ch;
schedule s;
   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 Enter your choice: ";
      cin>>ch;

      switch(ch)
      {
         case 1: s.scheduleEntry();
                break;

         case 2: s.display();
                break;

         case 3: s.createAppoint();
                break;

         case 4: s.cancelAppoint();
                break;

         case 5: s.sortAppoint();
                break;

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

      }
      cout<<"\n\n Do you want to continue? (y/n) : ";
      cin>>ans;
   }while(ans == 'y');
   return 0;
}
```