UCCD1004 PROGRAMMING CONCEPTS AND PRACTICES

ASSIGNMENT 2

	Member 1	Member 2
Name	Lee Jian Zhen	Jocelyn Tan Hui Ni
Student ID	19ACB02281	19ACB05937
Programme name	Bachelor of Computer Science	Bachelor of Computer Science
Group number	16	16

Objectives

Write a C++ program which enable an admin to book a ZOOM account (on behalf of lecturers) for a particular course to conduct online classes.

Pseudocode

Add:
Declare constant int size_1=60;
Declare courseName[size_1], b courseName[size_1] as char;
Declare numStudent, bnumStudent as int;
Decalre date,bdate as int;
Decalre zoom,bzoom as string;
Declare slot, bslot as string;
Declare bool reserved = false;
Ofstream outFile;

Date validation:
Declare void dateValidation(int&);
Declare int juneWeekend[4] = { 20,21,27,29 };
Declare int julyWeekend[8] = { 4,5,11,12,18,19,25,26 };
Declare int augWeekend[10] = { 1,2,8,9,15,16,22,23,29,30 };
Declare int septWeekend[6] = { 5,6,12,13,19,20 };
Declare int day;
Declare bool weekend, year, outOfRange;

Search:
Declare void Search();
Ifstream inFile;

Store:
Declare void Store (string,string,char*,int,int)

Check:
Declare void Check (bool&,string,string,char*,int,int);
Declare Reserved (bool&,string,int);

Update:
Declare void updateDetails (int,string,string);
Declare void update (string,string,char*,int,int);

Delete:
Declare void deleteSlot();
Ofstream temp;

Global Var:
Declare bool valid = false;
Declare t,n,select as int;

Declare menu {string line1, line2, line3, line4, line5, line6, line7} as struct;
Menu menuStart =
Display message
Welcome to Zoom booking services!
There are two Zoom accounts available: Zoom A and Zoom B.

Only two time slots, 11am ~ 1pm and 2pm ~ 4pm are available from Monday until Friday
The booking is just available for June trimester starting from 15 June 2020 to 20 September 2020.
Zoom A can fit in maximum 500 students per session while Zoom B is limited to 250 students per session.
Zoom A is used when Zoom B is not available or the class is having more than 250 students.

Do
n=0;
Reserved=false;
Valid=false;
Display message
Please select a services: [1] Book a slot [2] Search for booking details
[3] Update a booked course details(coursename / number of student).[4] Delete a booked slot.[5] Exit.
Ask user to select a service.
Do
If user input select,
If select >0 and <6, valid=true;
Else ask user to select from the list.
Else display message : It is not an integer! Please select again:;
While (!valid);
Valid=false;

If select=1;
Ask the user to input zoom, slot, courseName, numStudent,date;
If select=2;
Search();
If select=3;
Display message
Please insert the details of a slot you wish to update.
DateValidation(date);
Display message
Select a zoom slot: [1] Zoom A [2]Zoom B
Do
If user input select
If select >0 and<3
Valid=true;
Else display message:
Please select from above list
Else display message:
It is not an integer! Please select again:
While (!valid)
Valid=false;
If select =1;
Zoom= Zoom A;
Else
Zoom= Zoom B;

Display message
Select a time slot: [1] 11am-1pm [2] 2pm to 4pm
Do
If user input select
If select>0 and <3;
Valid=true;
Else display message:
Please select from above list
Else display message:
It is not an integer! Please select again:
While (!valid)
Valid=false;
If select=1;
Slot=11am to 1pm
Else
Slot=2pm to 4pm
updateDetails(date,zoom,slot)

If select=4;
deleteSlot();

If select !=5;
Display message: Thanks!

Function add:
Declare void Add (declare zoom,slot as string, courseName[size_1] as char, numStudent,date as int)
Ask user to insert course name;
Declare t as int and initialized to zero;
t<size_1;
Do increment of t;
Ask user to input number of students;
Do
If user input numStudent
If numStudent>0 and <501;
Valid=true;
Else display message: The number of students are out of range! Please insert again:
Else display message: It is not an integer! Please insert again:
While (!valid)
Valid=false;

If numStudent >250;
Display message: (Total number of students more than 250) Zoom A selected.
Zoom=Zoom A
Display message: Select a time slot: [1] 11am-1pm [2]2pm-4pm
Do
If user input select;
If select>0 and <3;
Valid true;
Else display message: Please select from above list:

Else display message: It is not an integer! Please select again:
While (!valid);
Valid=false;

If select=1;
Slot=11am to 1pm;
Check(reserved, zoom,slot,courseName,numStudent,date);
If it is not yet reserved ,store (zoom.slot,courseName,numStudent,date);
Else
Slot=2pm to 4pm;
Check(reserved, zoom,slot,courseName,numStudent,date);
If it is not yet reserved ,store (zoom.slot,courseName,numStudent,date);

Else reserved (reserved,zoom,date);
If reserved=true;
Display message:
(Total number of students less or equal to 250) (Zoom B is under reserved!) Zoom A is selected!
Zoom= Zoom A;
Display message: Select a time slot: [1] 11am to 1pm [2] 2pm to 4pm
Do
If user input select;
If select>0 and <3;
Valid true;
Else display message: Please select from above list:
Else display message: It is not an integer! Please select again:
While (!valid);
Valid=false;

If select =1;
Slot= 11am to 1pm;
Check(reserved, zoom,slot,courseName,numStudent,date);
If it is not yet reserved ,store (zoom.slot,courseName,numStudent,date);
Else
Slot=2pm to 4pm;
Check(reserved, zoom,slot,courseName,numStudent,date);
If it is not yet reserved ,store (zoom.slot,courseName,numStudent,date);

Else Display message:
(Total number of students less or equal to 250) Select a time slot: [1] Zoom A [2] Zoom B
Do
If user input select;
If select>0 and <3;
Valid true;
Else display message: Please select from above list:
Else display message: It is not an integer! Please select again:
While (!valid);
Valid=false;
If select =1;

Zoom = Zoom A;
Display message: Select a time slot: [1] 11am o 1pm [2] 2pm to 4pm
Do
If user input select;
If select>0 and <3;
Valid true;
Else display message: Please select from above list:
Else display message: It is not an integer! Please select again:
While (!valid);
Valid=false;
If select=1;
Slot=11am to 1pm;
Check(reserved, zoom,slot,courseName,numStudent,date);
If it is not yet reserved ,store (zoom.slot,courseName,numStudent,date);
Else
Slot=2pm to 4pm;
Check(reserved, zoom,slot,courseName,numStudent,date);
If it is not yet reserved ,store (zoom.slot,courseName,numStudent,date);

Else zoom=Zoom B
Display message: Select a time slot: [1] 11am o 1pm [2] 2pm to 4pm
Do
If user input select;
If select>0 and <3;
Valid true;
Else display message: Please select from above list:
Else display message: It is not an integer! Please select again:
While (!valid);
Valid=false;
If select=1;
Slot=11am to 1pm;
Check(reserved, zoom,slot,courseName,numStudent,date);
If it is not yet reserved ,store (zoom.slot,courseName,numStudent,date);
Else
Slot=2pm to 4pm;
Check(reserved, zoom,slot,courseName,numStudent,date);
If it is not yet reserved ,store (zoom.slot,courseName,numStudent,date);

Date validation:
Declare void date validation (int& date)
Display message : Insert a date (DDMMYYYY)>(15062020) (DD=Day, MM=Month, YYYY=Year)
Do
Weekend=false;
Year=false;
outOfRange=false;
Do
If user input date;
Valid=true;
Else display message: It is not an integer! Please insert again:

While(!valid)
Day= data/10000000;
If date >=10000000;
If date%10000 != 2020;
Year=true;
If date/10000%100=6;
t is initialized to zero;
t<4;
Do the increment of t;
If day<15 or >30;
outOfRange=true;
If day=juneWeekend[t];
Weekend=true;
If date/10000%100=7;
t is initialized to zero;
t<8;
Do the increment of t;
If day<15 or >31;
outOfRange=true;
If day=julyWeekend[t];
Weekend=true;
If date/10000%100=8;
t is initialized to zero;
t<10;
Do the increment of t;
If day<15 or >31;
outOfRange=true;
If day=augWeekend[t];
Weekend=true;
If date/10000%100=9;
t is initialized to zero;
t<6;
Do the increment of t;
If day<15 or >20;
outOfRange=true;
If day=septWeekend[t];
Weekend=true;
Else outOfRange=true;

If date<10000000;
If date%10000 != 2020;
Year=true;
If date/10000%10=6;
t is initialized to zero;
t<4;
Do the increment of t;
If day<15 or >30;
outOfRange=true;
If day=juneWeekend[t];
Weekend=true;

If date/10000%10=7;
t is initialized to zero;
t<8;
Do the increment of t;
If day<15 or >31;
outOfRange=true;
If day=julyWeekend[t];
Weekend=true;
If date/10000%10=8;
t is initialized to zero;
t<10;
Do the increment of t;
If day<15 or >31;
outOfRange=true;
If day=augWeekend[t];
Weekend=true;
If date/10000%10=9;
t is initialized to zero;
t<6;
Do the increment of t;
If day<15 or >20;
outOfRange=true;
If day=septWeekend[t];
Weekend=true;
Else outOfRange=true;
If year=true;
Display message: It is not 2020! Please insert again:
If weekend=true;
Display message: It is weekend! Please insert again:
If outOfRange=true;
Display message: The date is out of range! Please insert again:
While weekend or year or outOfRange = true;

Check reserved or not for update or keep
Declare void check
(declare reseved as bool&, zoom,slot as string, courseName[size_1] as char,
numStudent,date as int)
If bzoom= “”;
Break;
If date=bdate;
If zoom=bzoom or slot=bslot;
Display the reserved detail no, zoom,slot,course name, number of student and date;
Do the increment of n;
If reserved;
Display message: The slot had booked by the above details.
Ask whether they want to update or keep;
Do
If user input select;
Select>0 or <3;
Valid=true;

Else ask the user to input from the above list;
Else ask the user to input again;
If select=1;
Update (zoom,slot,couorseName, numStudent,date)
Else display message: Keep successfully!;

Store information into booking.txt:
Declare void store
(declare zoom,slot as string, courseName[size_1]as char, numStudent ,date as int);
Store the booking details;
Display message: Book successfully!;

Check whether Zoom B had be booked or not:
Declare void reserved
(declare reserved as bool&, zoom as string, date as int)
If bzoom="";
Break;
If date=bdate;
Reserved=true;

Search:
Declare void Search();
T=0;
Display message:
Search by: [1] Zoom slot [2] Time slot [3] Course name [4] Date
Do
If user input select;
Select>0 and <5;
Valid=true;
Else ask the user to input from the above list;
Else ask the user to input again;
Valid=false;

If select=1;
Display message:
Select a Zoom slot: [1] Zoom A [2] Zoom B
Do
If user input select;
Select>0 and <3;
Valid=true;
Else ask the user to input from the above list;
Else ask the user to input again;
Valid=false;

If select =1;
Zoom= Zoom A;
Else
Zoom = Zoom B;
If inFile is open;
If bzoom= "";

Break;
If zoom=bzoom;
Display the booking details;
Do the increment of n;
Reserved= true;
If reserved;
Display message: Above are the booked details!
Else display message: No booked details!
Reserved =false;
Else display message: Unable to open file!

If select=2;
Display message:
Select a time slot: [1] 11am to 1pm [2] 2pm to 4pm
Do
If user input select;
Select>0 and <3;
Valid=true;
Else ask the user to input from the above list;
Else ask the user to input again;
Valid=false;

If select =1;
Slot= 11am to 1pm;
Else
Slot= 2om to 4pm;
If inFile is open;
If bzoom= “”;
Break;
If zoom=bzoom;
Display the booking details;
Do the increment of n;
Reserved= true;
If reserved;
Display message: Above are the booked details!
Else display message: No booked details!
Reserved =false;
Else display message: Unable to open file!

If select=3;
Display message: Insert course name;
t is initialised to zero;
t<size_1;
Do the increment of t;
If inFile is open;
If bzoom= “”;
Break;
If zoom=bzoom;
Display the booking details;
Do the increment of n;

Reserved =true;
If reserved;
Display message: Above are the booked details!
Else display message: No booked details!
Reserved= false;
Else display message: Unable to open file!

If select =4;
dateValidation(date);
If inFile is open;
If bzoom= “”;
Break;
If zoom=bzoom;
Display the booking details;
Do the increment of n;
Reserved= true;
If reserved;
Display message: Above are the booked details!
Else display message: No booked details!
Reserved= false;
Else display message: Unable to open file!

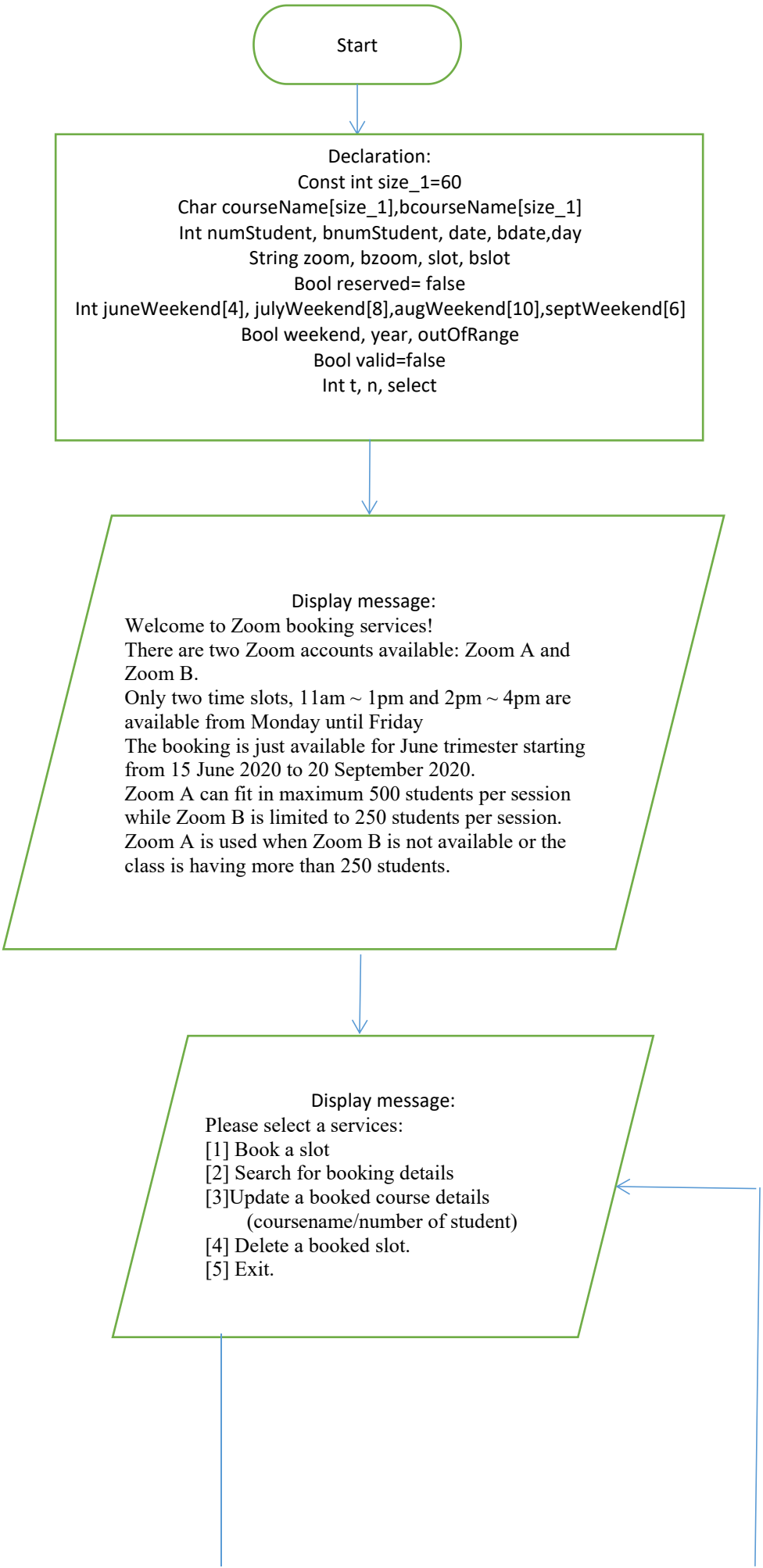
Update:
Declare void updateDetails
(date as int, zoom, slot as string);
If inFile is open;
If bzoom= “”;
Break;
If date=bdate;
If zoom=bzoom and slot=bslot;
Display the booking details;
Ask the user to input new course name;
t is initialised to zero;
t<size_1;
Do the increment of t;
Ask the user to input the new number of students;
Do
If zoom= Zoom A;
If user input numStudent;
If numStudent>0 and <501;
Valid=true;
Else display message: The number of students are out of range! Please insert again:;
Else display message: It is not an integer! Please insert again:;
Else if user input numStudent;
numStudent>0 and <250;
Valid=true;
Else display message: The number of students are out of range! Please insert again:;
Else display message: It is not an integer! Please insert again:;
Valid=false;

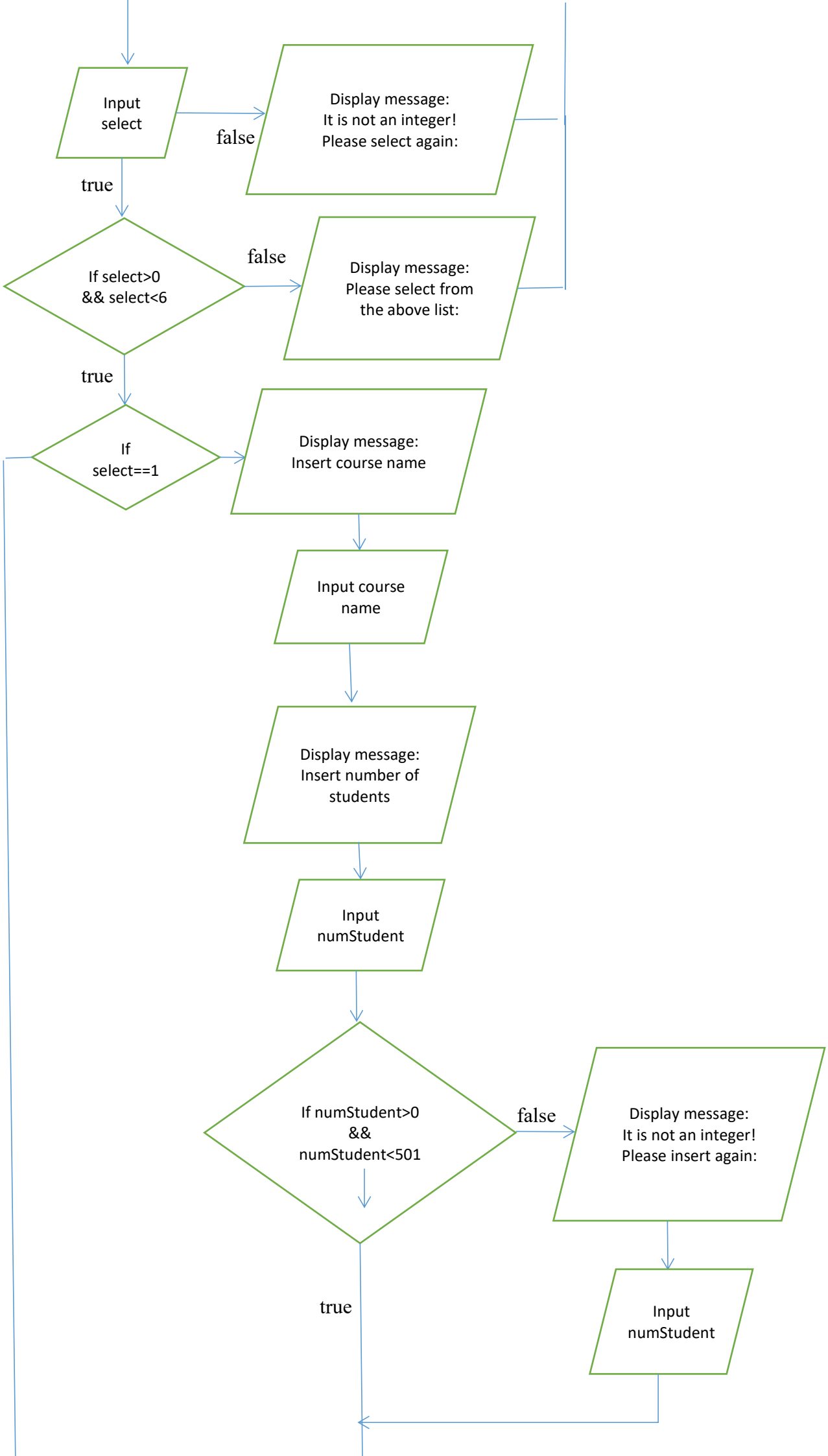
If bzoom= “”;
Break;
If date!=bdate or slot !=bslot;
Display the booking details;
Display message: The details has been deleted successfully!;
Remove the previous booking from booking.txt
Else display message: Please just insert the details of ONE slot.
Else display message: There is no booked slot.
Else display message: Unable to open file!

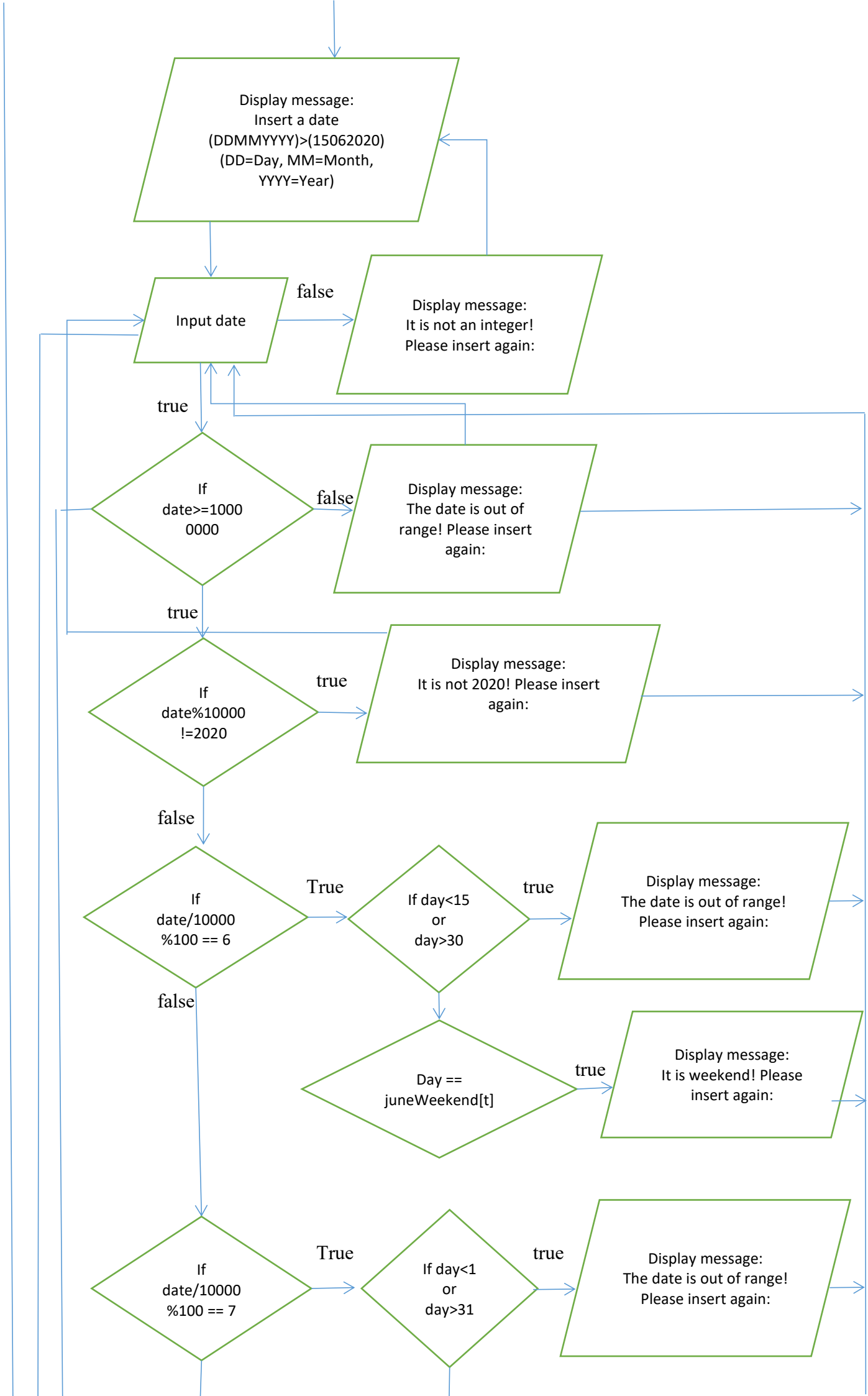
Declare void update
(Declare zoom, slot as string, courseName[size_1] as char, numStudent,date as int);
If inFile is open;
If bzoom= “”;
Break;
If date=bdate;
If zoom=bzoom or slot=bslot;
Continue;
Update the new booking details;
Display message: The booking has been updated successfully!
Remove the previous booking details from booking.txt;

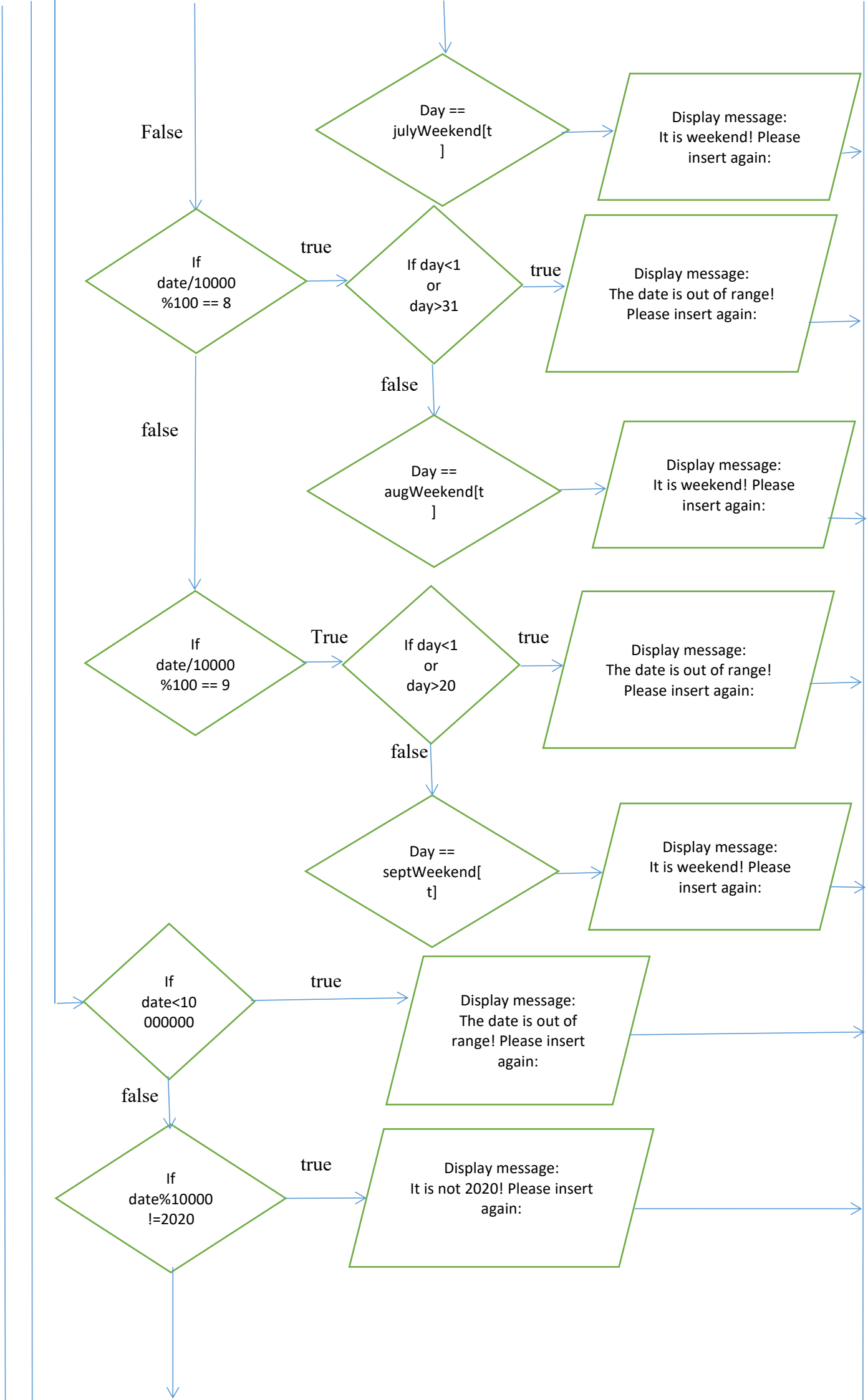
Delete:
Declare void deleteSlot;
Ask the user to input the course name they wish to delete;
t is initialized to zero;
t<size_1;
Do the increment of t;
If inFile is open;
If bzoom= “”;
Break;
If string compare courseName and bcourseName !=0;
Store the booking details;
Else display the deleted booking details;
Display message: The booking has been deleted successfully!;
The booking details are removed from booking.txt;
Else display message: Unable to open file!;

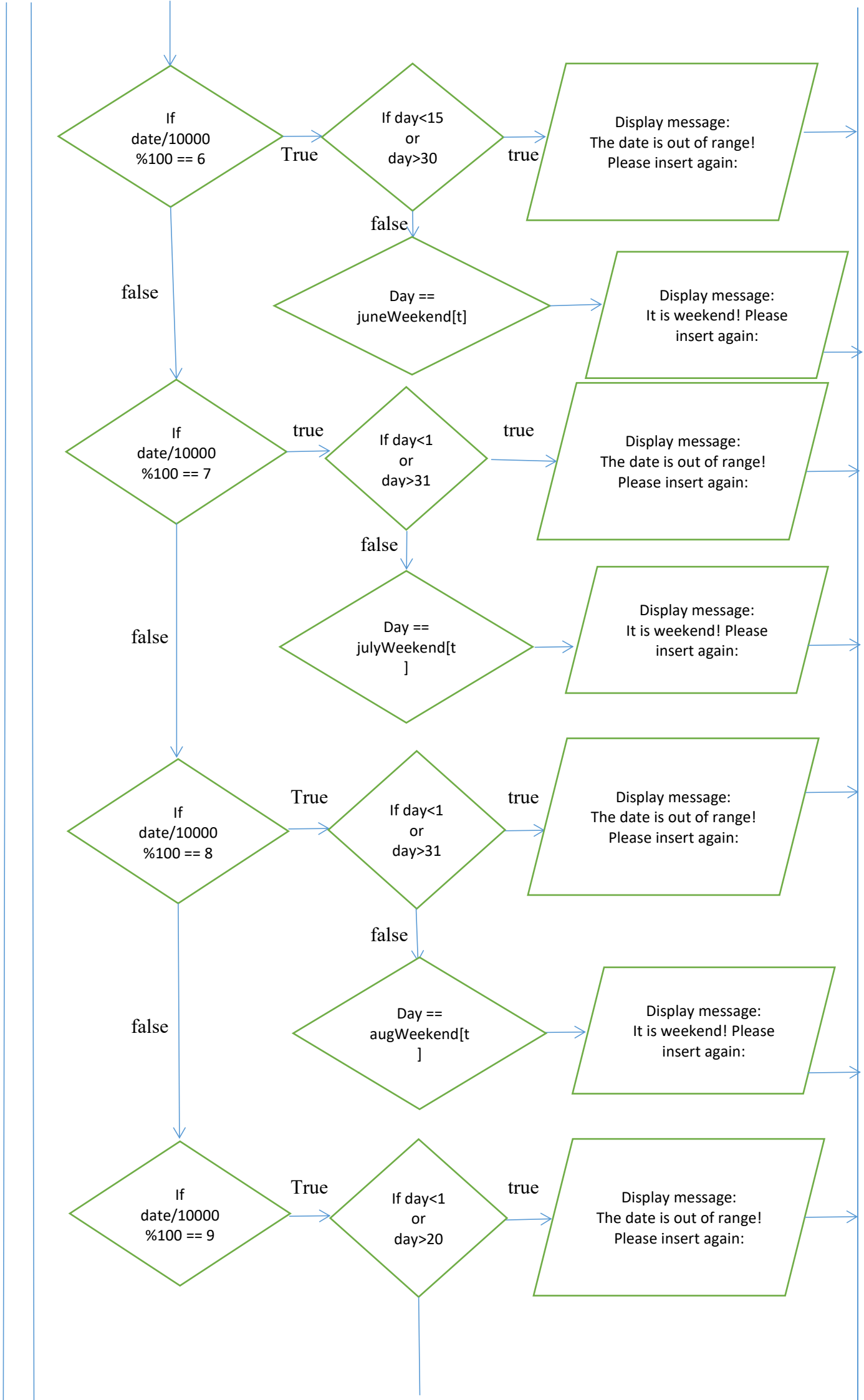
Flowchart

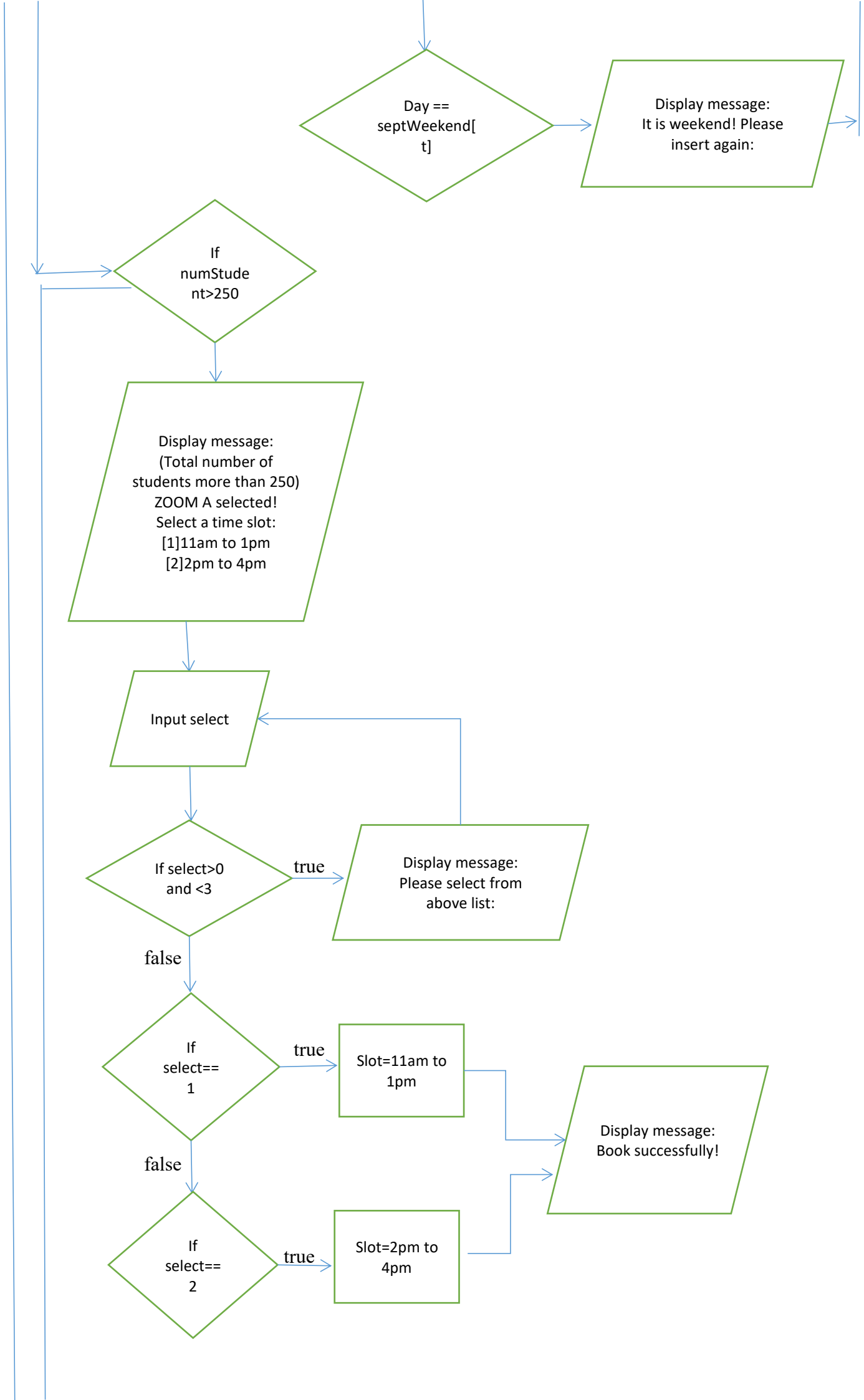

Start
Declaration:
Const int size_1=60
Char courseName[size_1],bcourseName[size_1]
Int numStudent, bnumStudent, date, bdate,day
String zoom, bzoom, slot, bslot
Bool reserved= false
Int juneWeekend[4], julyWeekend[8],augWeekend[10],septWeekend[6]
Bool weekend, year, outOfRange
Bool valid=false
Int t, n, select
Display message:
Welcome to Zoom booking services!
There are two Zoom accounts available: Zoom A and Zoom B.
Only two time slots, 11am ~ 1pm and 2pm ~ 4pm are available from Monday until Friday
The booking is just available for June trimester starting from 15 June 2020 to 20 September 2020.
Zoom A can fit in maximum 500 students per session while Zoom B is limited to 250 students per session.
Zoom A is used when Zoom B is not available or the class is having more than 250 students.
Display message:
Please select a services:
[1] Book a slot
[2] Search for booking details
[3]Update a booked course details
(coursename/number of student)
[4] Delete a booked slot.
[5] Exit.

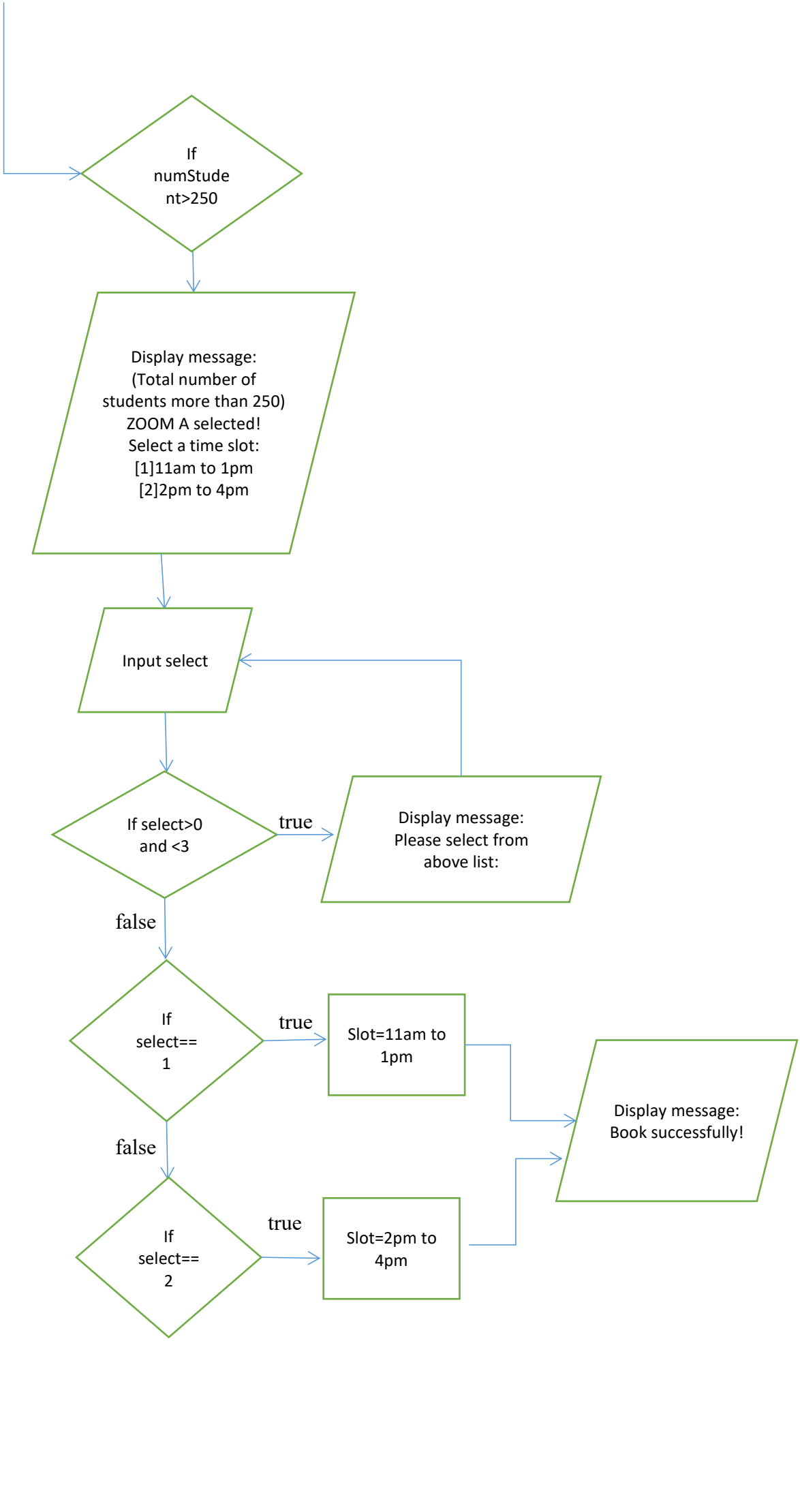

Input select
false
Display message: It is not an integer! Please select again:
true
If select>0 && select<6
false
Display message: Please select from the above list:
true
If select==1
Display message: Insert course name
Input course name
Display message: Insert number of students
Input numStudent
If numStudent>0 && numStudent<501
false
Display message: It is not an integer! Please insert again:
Input numStudent
true

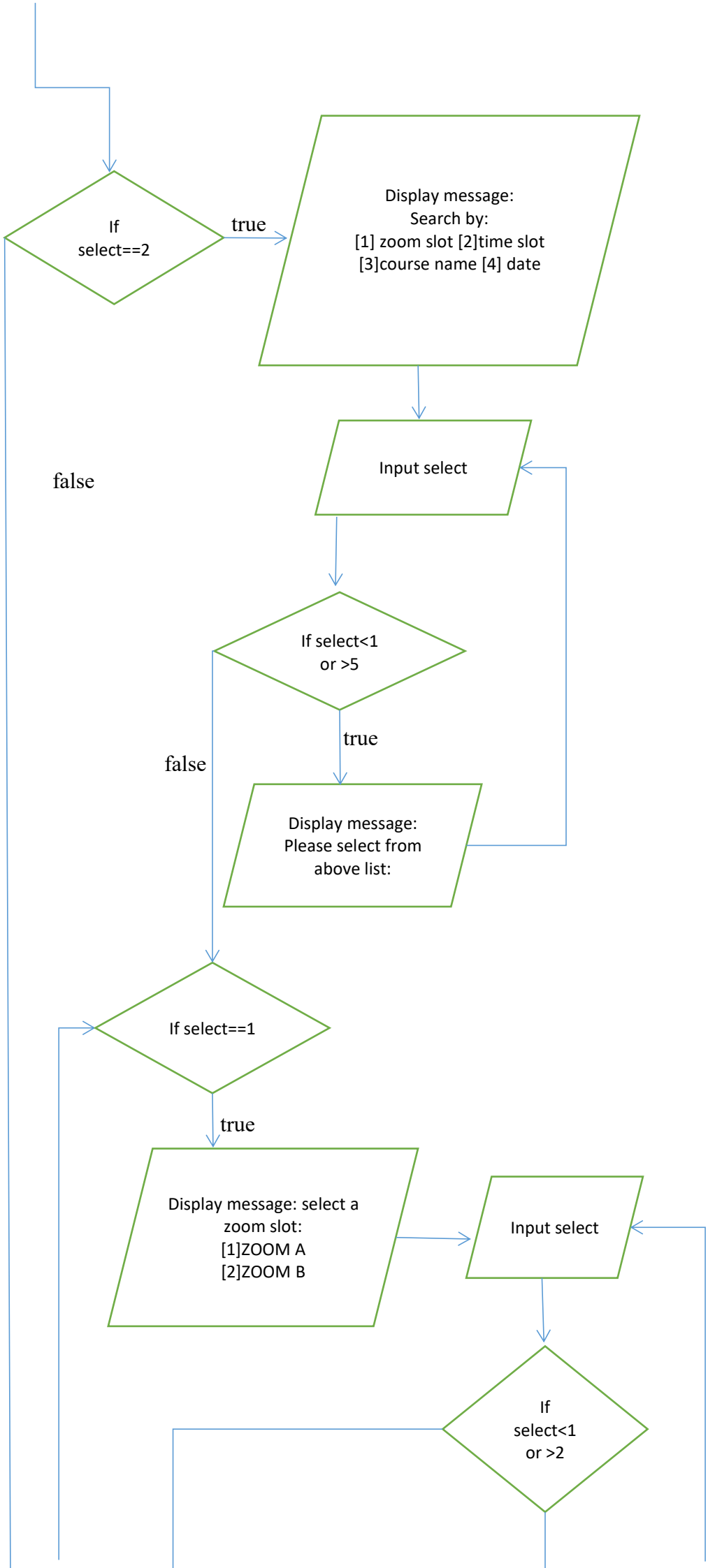

Display message:
Insert a date
(DDMMYYYY)>(15062020)
(DD=Day, MM=Month,
YYYY=Year)
Input date
false
Display message:
It is not an integer!
Please insert again:
true
If
date>=1000
0000
false
Display message:
The date is out of
range! Please insert
again:
true
If
date%10000
!=2020
true
Display message:
It is not 2020! Please insert
again:
false
If
date/10000
%100 == 6
True
If day<15
or
day>30
true
Display message:
The date is out of range!
Please insert again:
false
Day ==
juneWeekend[t]
true
Display message:
It is weekend! Please
insert again:
If
date/10000
%100 == 7
True
If day<1
or
day>31
true
Display message:
The date is out of range!
Please insert again:

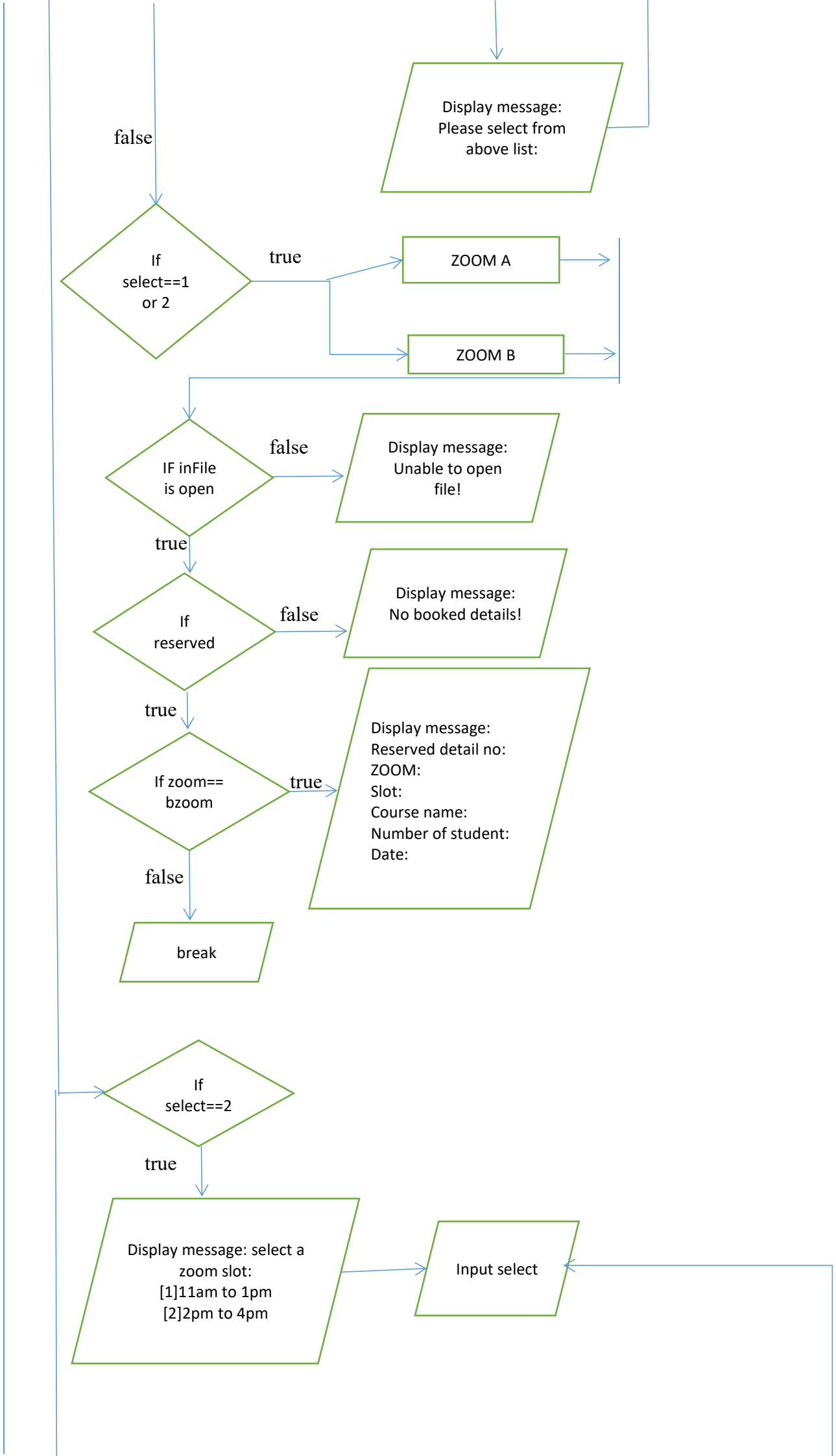

Day == julyWeekend[t]
Display message: It is weekend! Please insert again:
False
If date/10000 %100 == 8
true
If day<1 or day>31
true
Display message: The date is out of range! Please insert again:
false
false
Day == augWeekend[t]
Display message: It is weekend! Please insert again:
If date/10000 %100 == 9
True
If day<1 or day>20
true
Display message: The date is out of range! Please insert again:
false
Day == septWeekend[t]
Display message: It is weekend! Please insert again:
If date<10 000000
true
Display message: The date is out of range! Please insert again:
false
If date%10000 !=2020
true
Display message: It is not 2020! Please insert again:

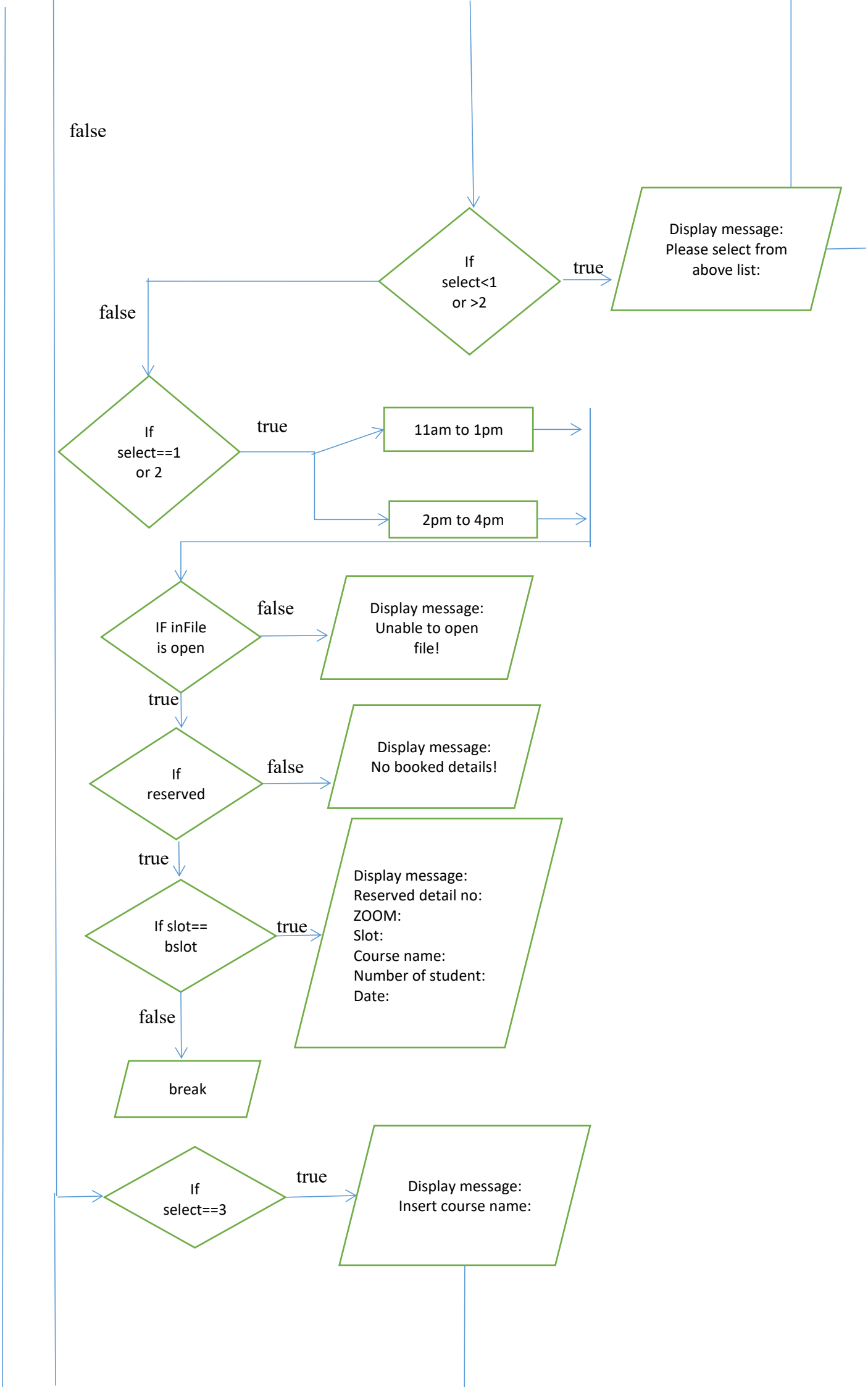

If
date/10000
%100 == 6
True
If day<15
or
day>30
true
Display message:
The date is out of range!
Please insert again:
false
false
Day ==
juneWeekend[t]
Display message:
It is weekend! Please
insert again:
If
date/10000
%100 == 7
true
If day<1
or
day>31
true
Display message:
The date is out of range!
Please insert again:
false
false
Day ==
julyWeekend[t
]
Display message:
It is weekend! Please
insert again:
If
date/10000
%100 == 8
True
If day<1
or
day>31
true
Display message:
The date is out of range!
Please insert again:
false
false
Day ==
augWeekend[t
]
Display message:
It is weekend! Please
insert again:
If
date/10000
%100 == 9
True
If day<1
or
day>20
true
Display message:
The date is out of range!
Please insert again:

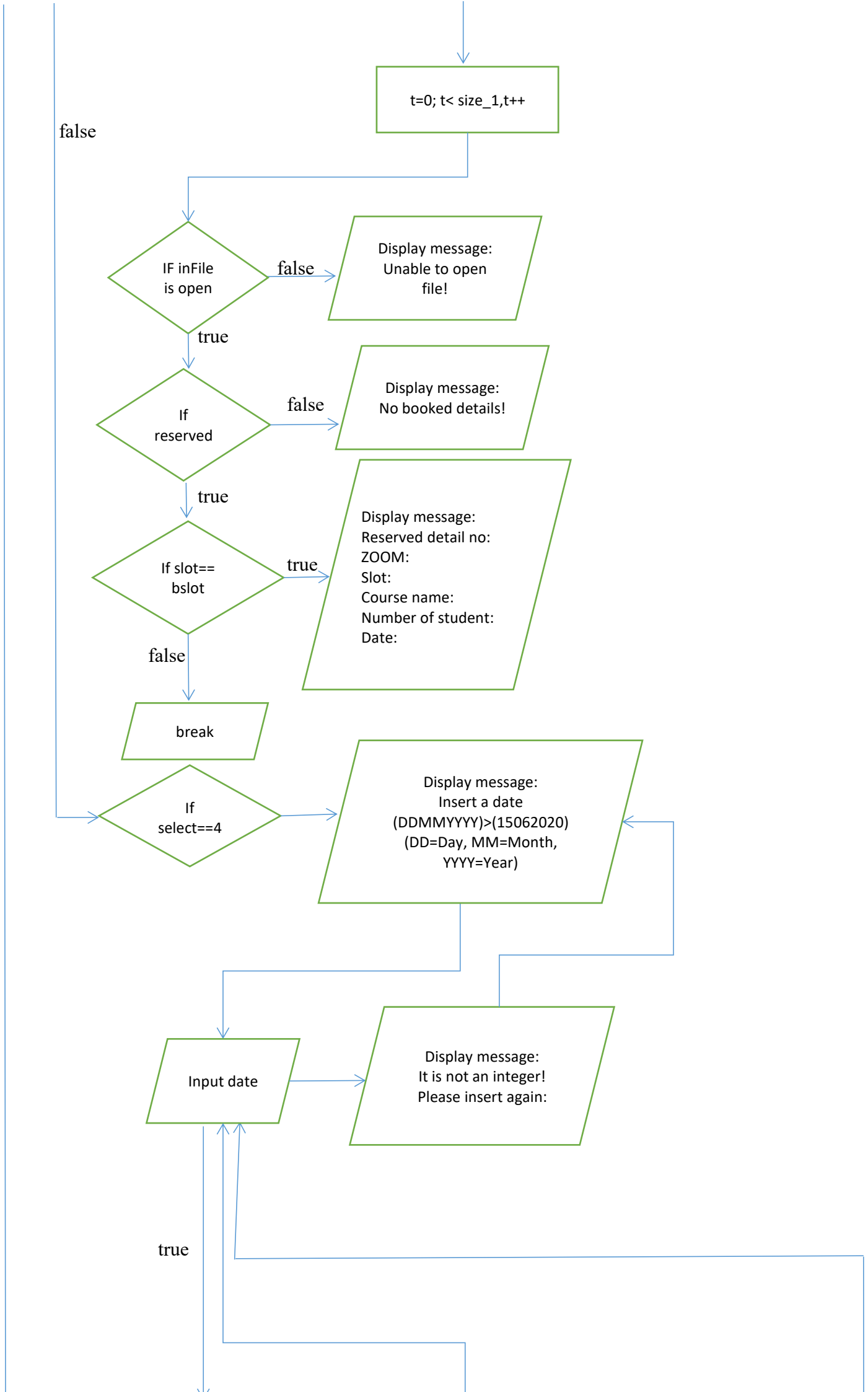

Day == septWeekend[t]
Display message: It is weekend! Please insert again:
If numStude nt>250
Display message: (Total number of students more than 250) ZOOM A selected! Select a time slot: [1]11am to 1pm [2]2pm to 4pm
Input select
If select>0 and <3
true
Display message: Please select from above list:
false
If select== 1
true
Slot=11am to 1pm
Display message: Book successfully!
false
If select== 2
true
Slot=2pm to 4pm

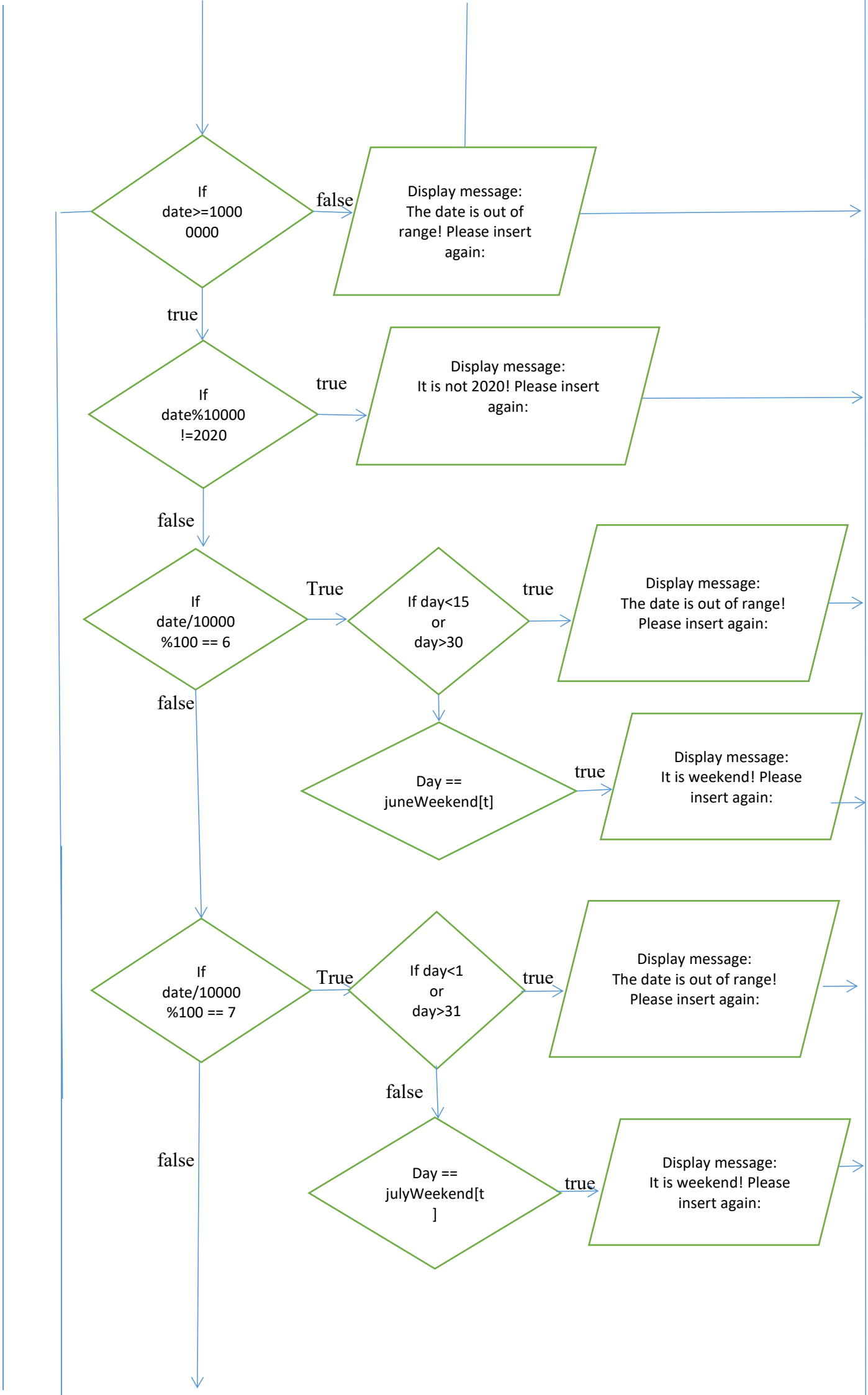

If numStudent>250
Display message: (Total number of students more than 250) ZOOM A selected! Select a time slot: [1]11am to 1pm [2]2pm to 4pm
Input select
If select>0 and <3
true
Display message: Please select from above list:
false
If select== 1
true
Slot=11am to 1pm
false
If select== 2
true
Slot=2pm to 4pm
Display message: Book successfully!

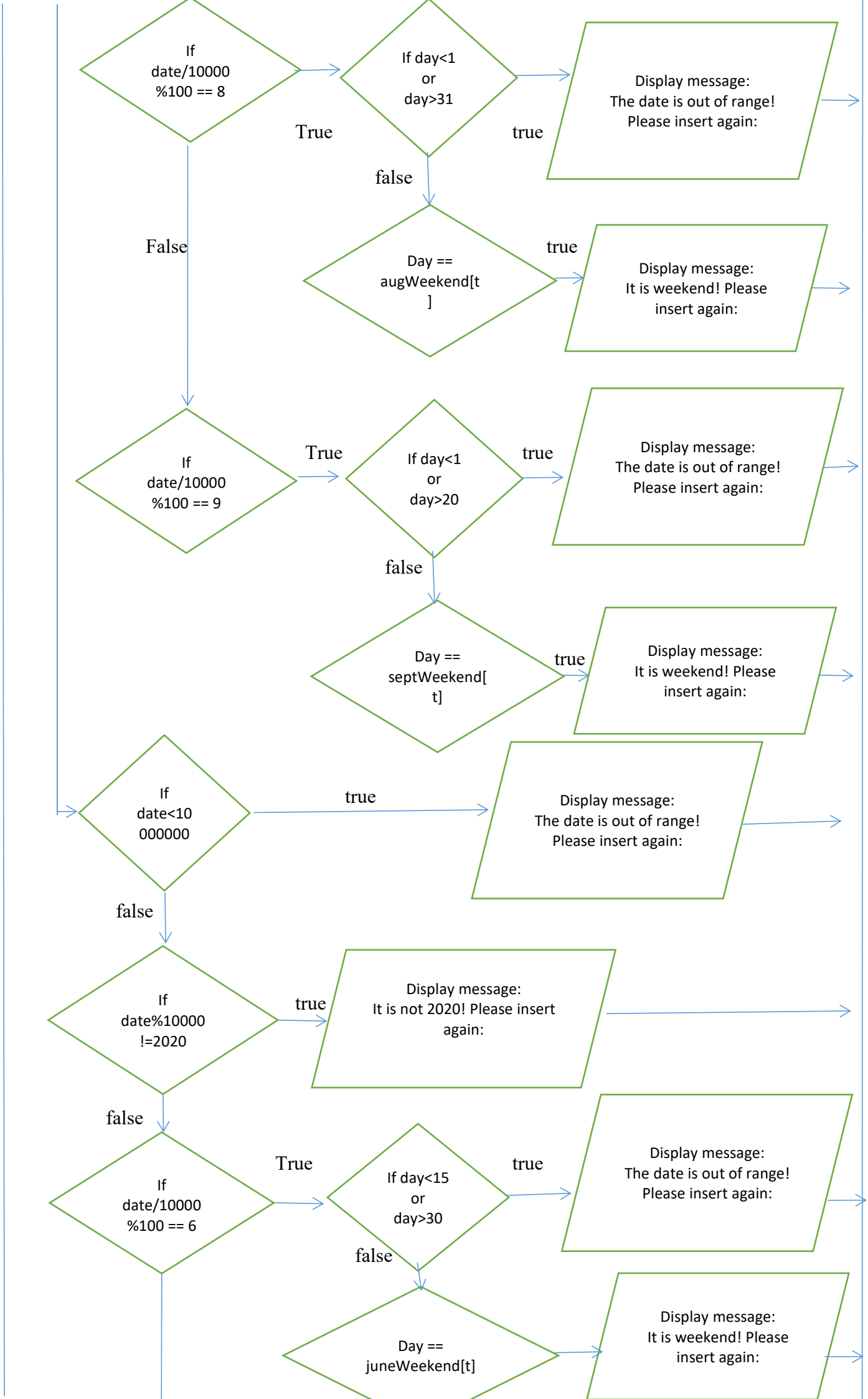

If
select==2
true
Display message:
Search by:
[1] zoom slot [2]time slot
[3]course name [4] date
Input select
false
If select<1
or >5
true
false
Display message:
Please select from
above list:
If select==1
true
Display message: select a
zoom slot:
[1]ZOOM A
[2]ZOOM B
Input select
If
select<1
or >2

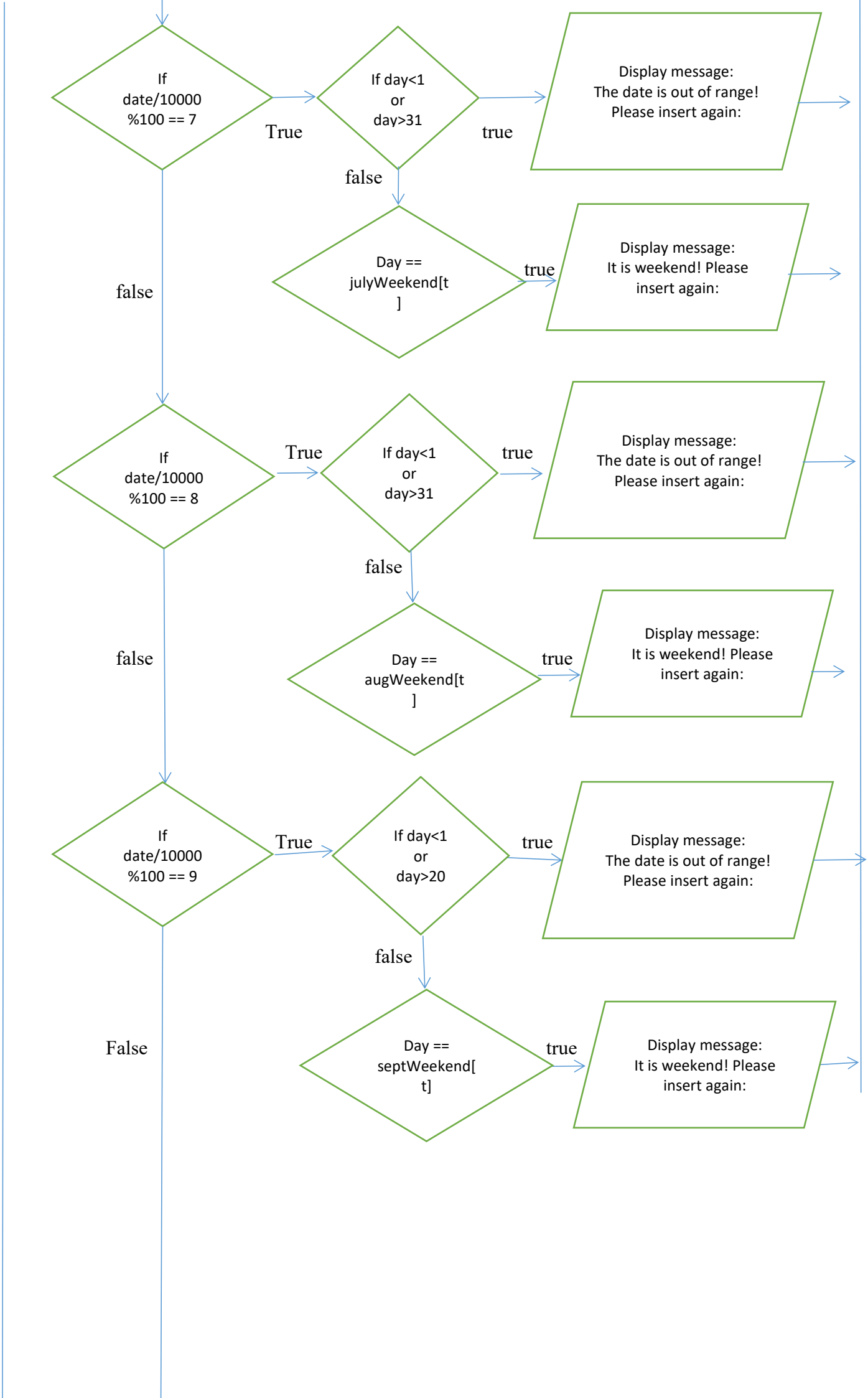

Display message:
Please select from
above list:
false
If
select==1
or 2
true
ZOOM A
ZOOM B
IF inFile
is open
false
Display message:
Unable to open
file!
true
If
reserved
false
Display message:
No booked details!
true
If zoom==
bzoom
true
Display message:
Reserved detail no:
ZOOM:
Slot:
Course name:
Number of student:
Date:
false
break
If
select==2
true
Display message: select a
zoom slot:
[1]11am to 1pm
[2]2pm to 4pm
Input select

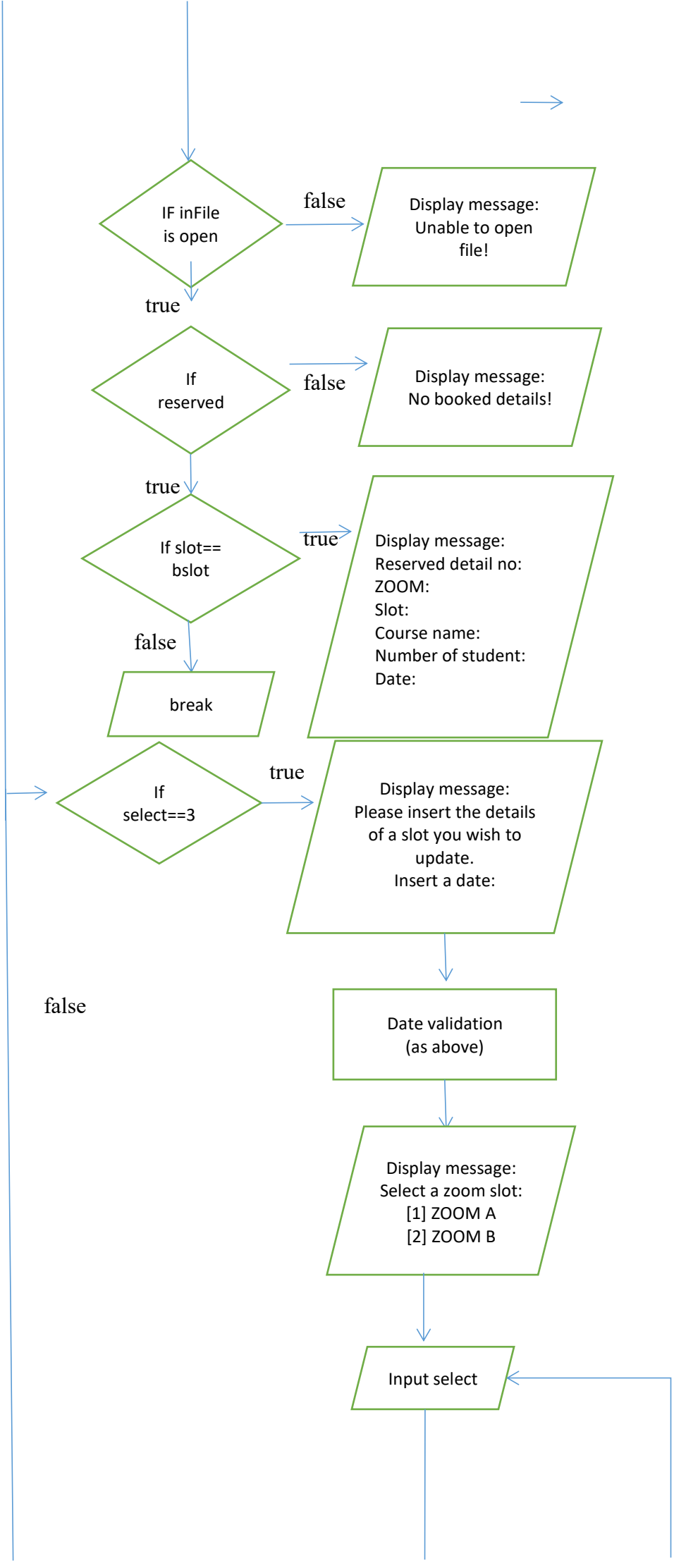

false
If
select<1
or >2
true
Display message:
Please select from
above list:
false
If
select==1
or 2
true
11am to 1pm
2pm to 4pm
IF inFile
is open
false
Display message:
Unable to open
file!
true
If
reserved
false
Display message:
No booked details!
true
If slot==
bslot
true
Display message:
Reserved detail no:
ZOOM:
Slot:
Course name:
Number of student:
Date:
false
break
If
select==3
true
Display message:
Insert course name:

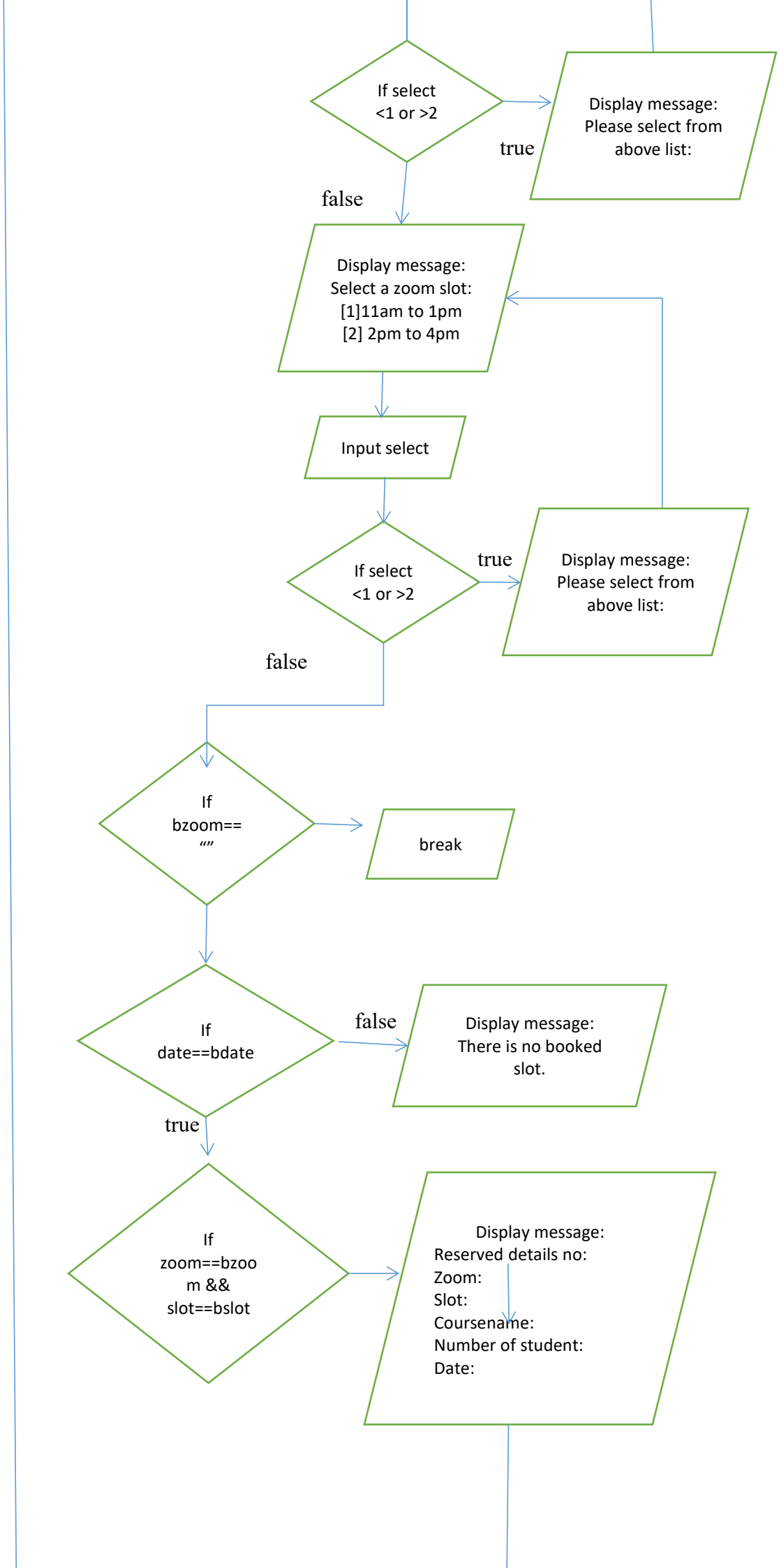

t=0; t< size_1,t++
false
IF inFile is open
false
Display message: Unable to open file!
true
If reserved
false
Display message: No booked details!
true
If slot== bslot
true
Display message: Reserved detail no: ZOOM: Slot: Course name: Number of student: Date:
false
break
If select==4
Display message: Insert a date (DDMMYYYY)>(15062020) (DD=Day, MM=Month, YYYY=Year)
Input date
Display message: It is not an integer! Please insert again:
true

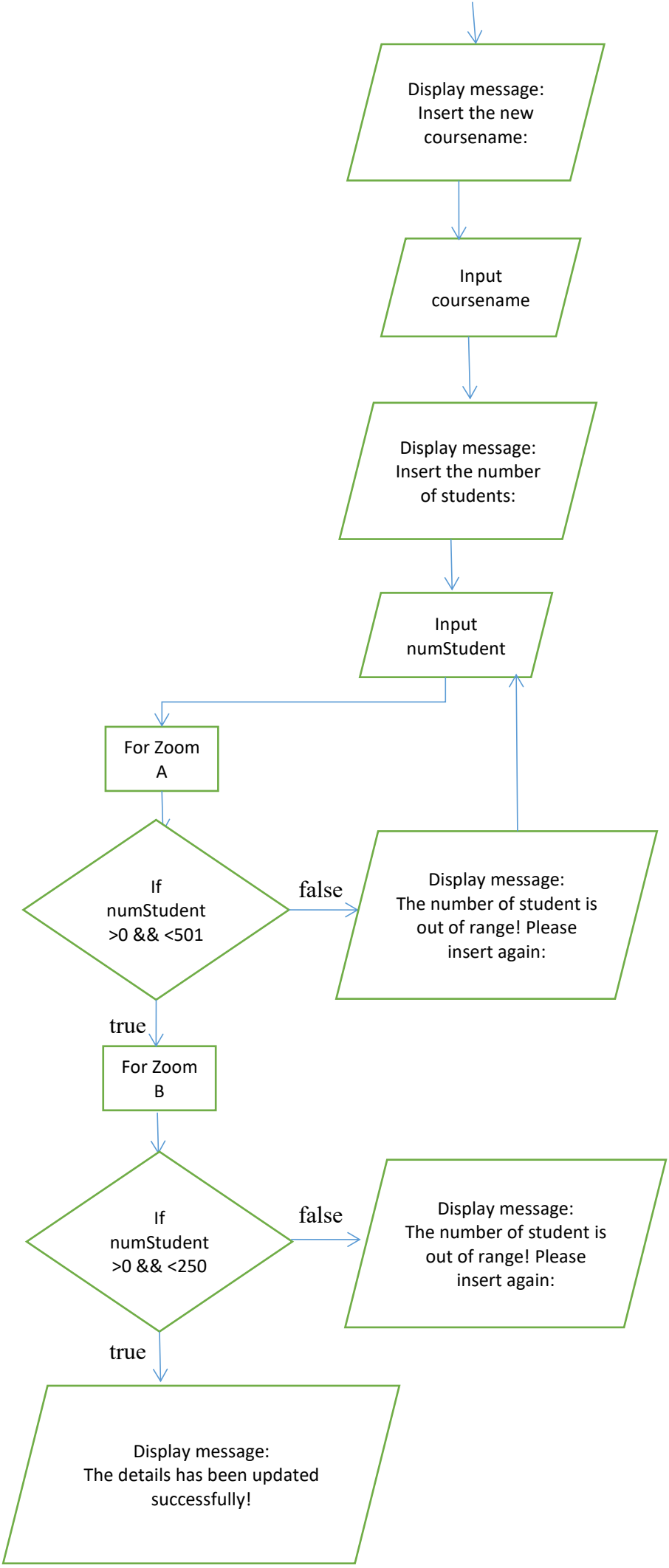

If date>=10000000
false
Display message: The date is out of range! Please insert again:
true
If date%10000 !=2020
true
Display message: It is not 2020! Please insert again:
false
If date/10000 %100 == 6
True
If day<15 or day>30
true
Display message: The date is out of range! Please insert again:
false
Day == juneWeekend[t]
true
Display message: It is weekend! Please insert again:
If date/10000 %100 == 7
True
If day<1 or day>31
true
Display message: The date is out of range! Please insert again:
false
false
Day == julyWeekend[t]
true
Display message: It is weekend! Please insert again:

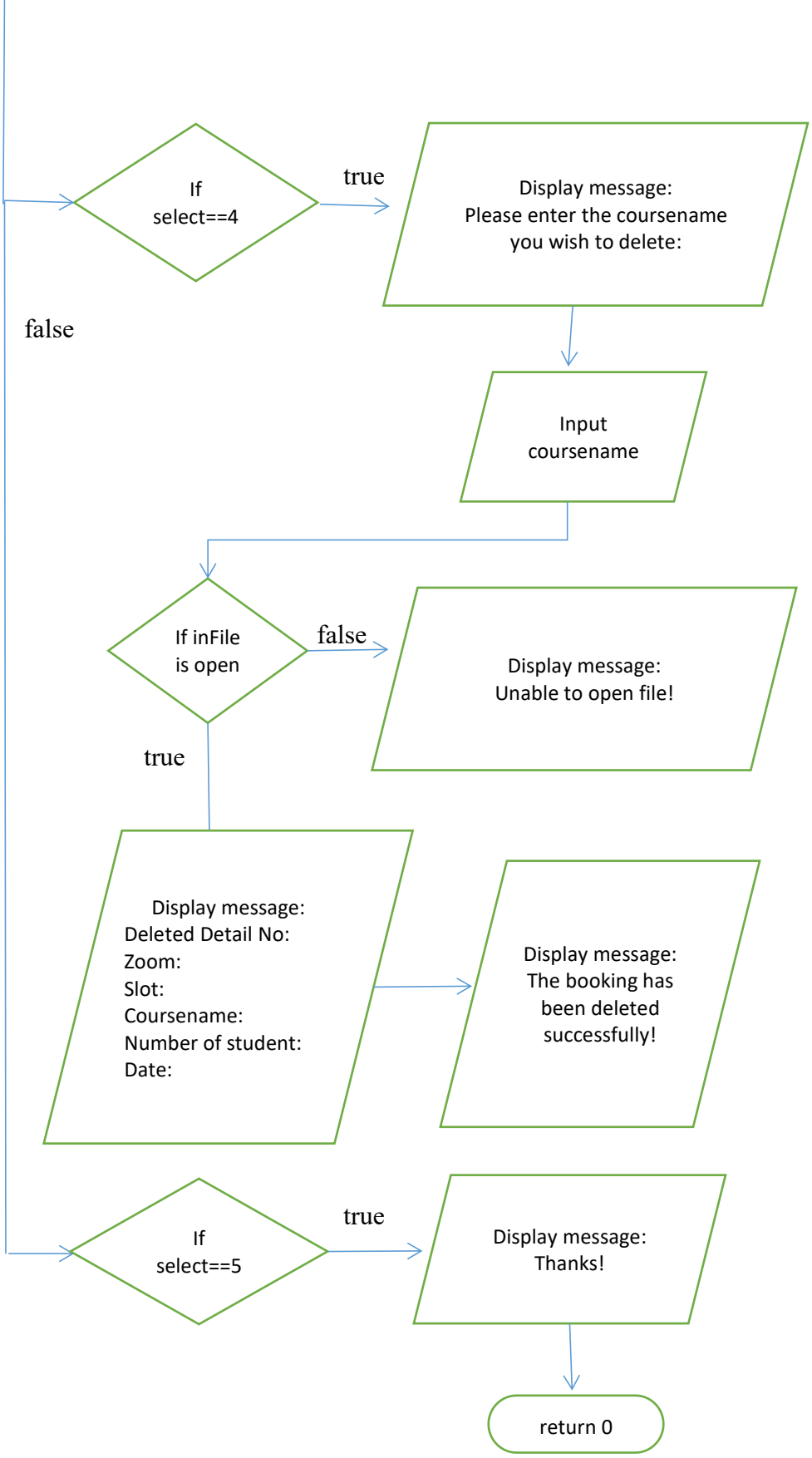

If date/10000 %100 == 8
True
If day<1 or day>31
true
Display message: The date is out of range! Please insert again:
false
False
Day == augWeekend[t]
true
Display message: It is weekend! Please insert again:
If date/10000 %100 == 9
True
If day<1 or day>20
true
Display message: The date is out of range! Please insert again:
false
Day == septWeekend[t]
true
Display message: It is weekend! Please insert again:
If date<10 000000
true
Display message: The date is out of range! Please insert again:
false
If date%10000 !=2020
true
Display message: It is not 2020! Please insert again:
false
If date/10000 %100 == 6
True
If day<15 or day>30
true
Display message: The date is out of range! Please insert again:
false
Day == juneWeekend[t]
Display message: It is weekend! Please insert again:


If
date/10000
%100 == 7
True
If day<1
or
day>31
true
Display message:
The date is out of range!
Please insert again:
false
Day ==
julyWeekend[t
]
true
Display message:
It is weekend! Please
insert again:
false
If
date/10000
%100 == 8
True
If day<1
or
day>31
true
Display message:
The date is out of range!
Please insert again:
false
Day ==
augWeekend[t
]
true
Display message:
It is weekend! Please
insert again:
false
If
date/10000
%100 == 9
True
If day<1
or
day>20
true
Display message:
The date is out of range!
Please insert again:
false
Day ==
septWeekend[
t]
true
Display message:
It is weekend! Please
insert again:
False

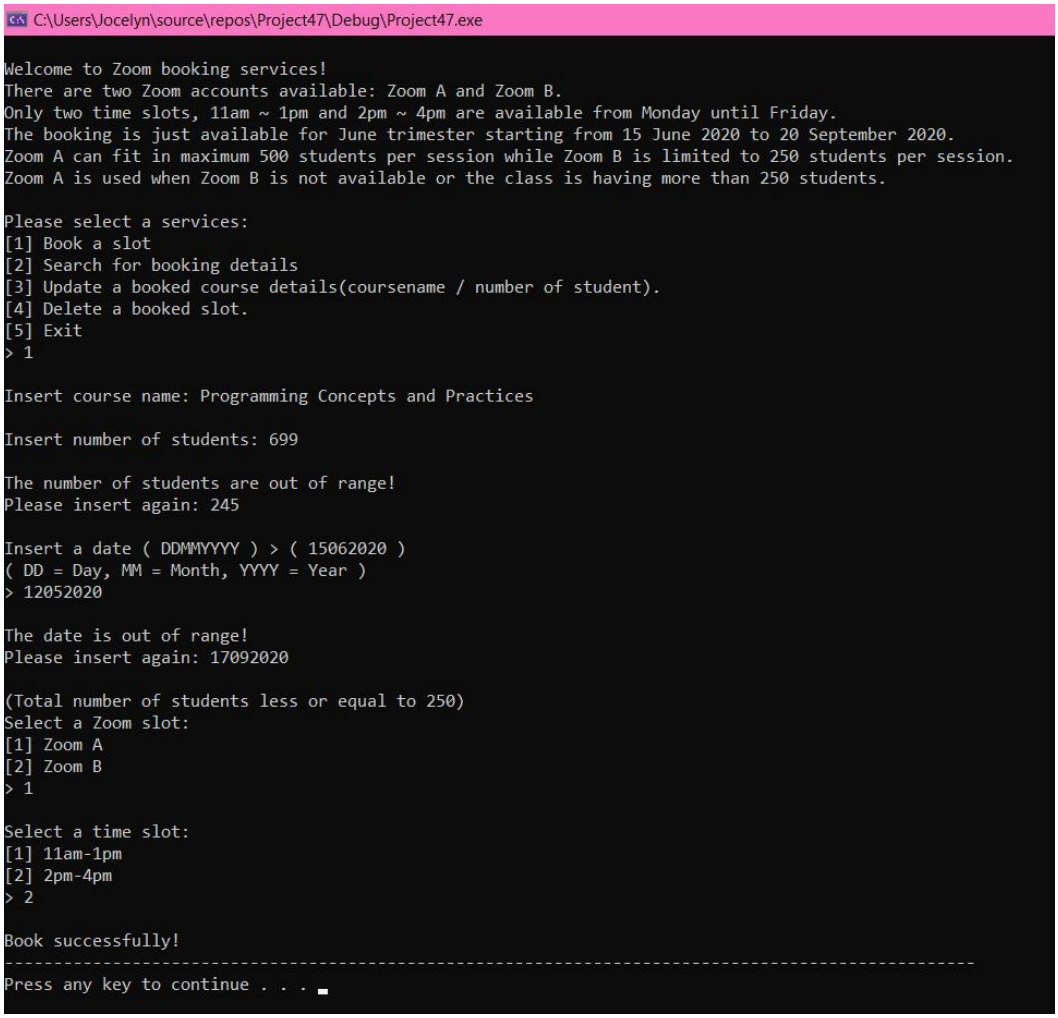

IF inFile is open
false
Display message: Unable to open file!
true
If reserved
false
Display message: No booked details!
true
If slot== bslot
true
Display message: Reserved detail no: ZOOM: Slot: Course name: Number of student: Date:
false
break
If select==3
true
Display message: Please insert the details of a slot you wish to update. Insert a date:
false
Date validation (as above)
Display message: Select a zoom slot: [1] ZOOM A [2] ZOOM B
Input select

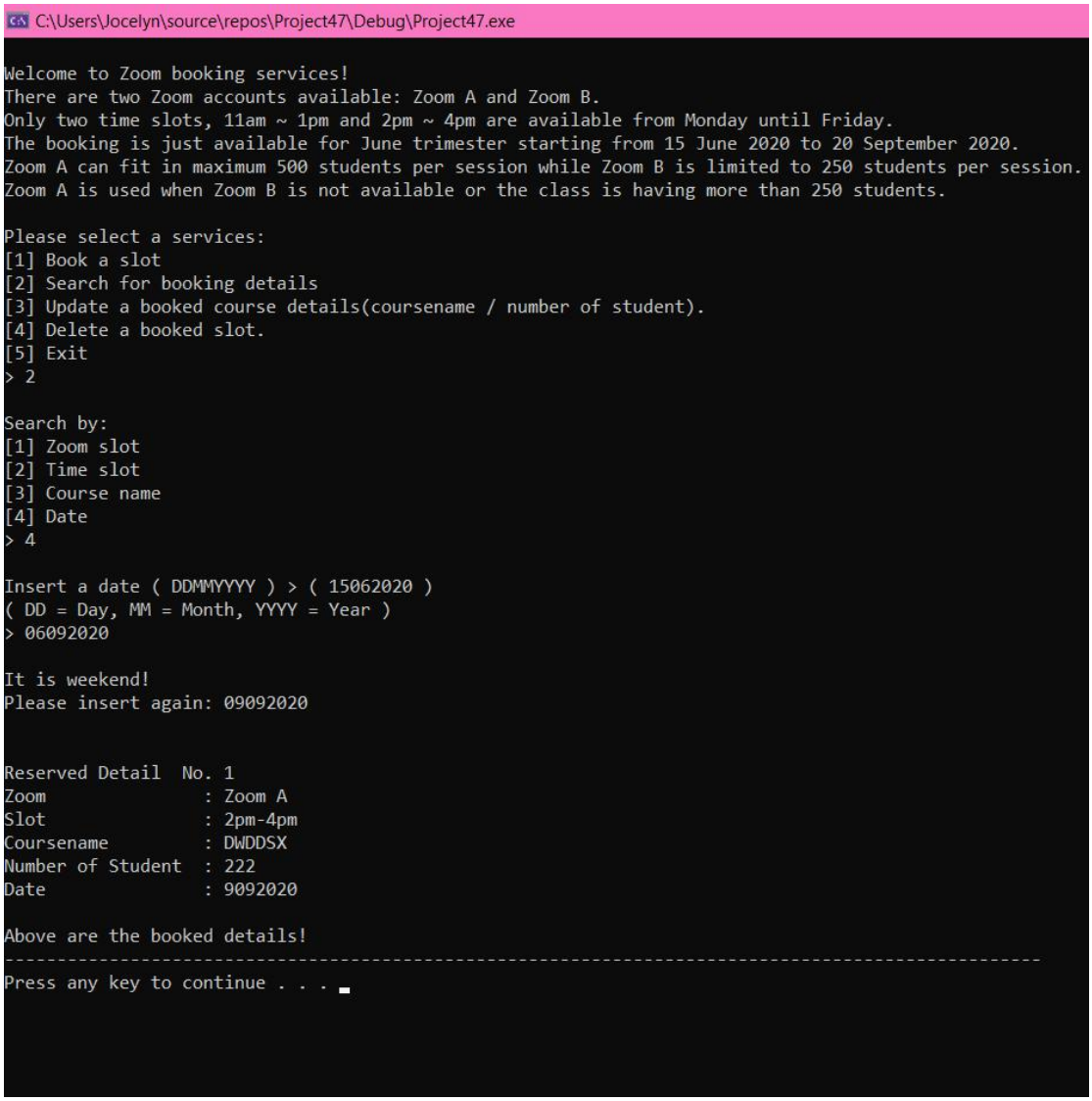

If select <1 or >2
true
Display message: Please select from above list:
false
Display message: Select a zoom slot: [1]11am to 1pm [2] 2pm to 4pm
Input select
If select <1 or >2
true
Display message: Please select from above list:
false
If bzoom== “”
break
If date==bdate
false
Display message: There is no booked slot.
true
If zoom==bzoom && slot==bslot
Display message: Reserved details no: Zoom: Slot: Coursename: Number of student: Date:

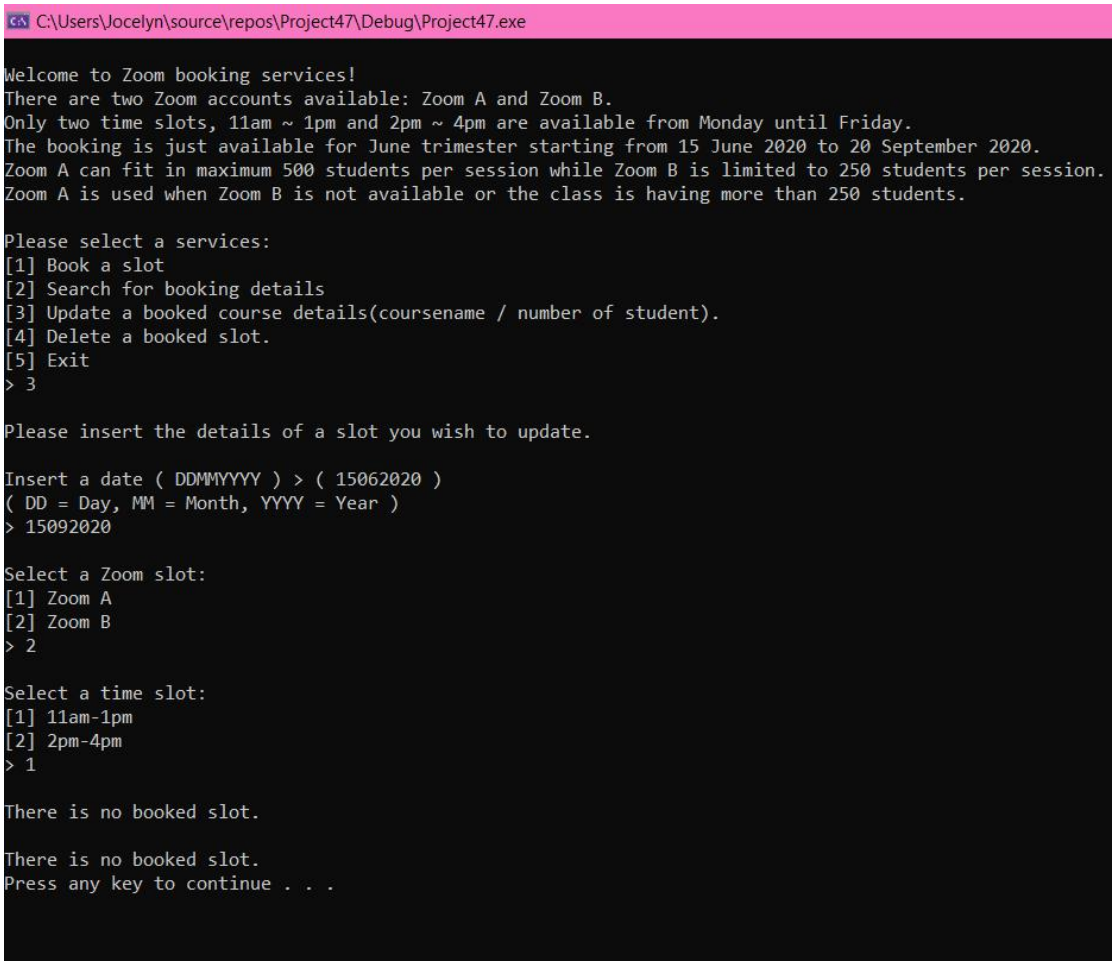

Display message:
Insert the new
coursename:
Input
coursename
Display message:
Insert the number
of students:
Input
numStudent
For Zoom
A
If
numStudent
>0 && <501
false
Display message:
The number of student is
out of range! Please
insert again:
true
For Zoom
B
If
numStudent
>0 && <250
false
Display message:
The number of student is
out of range! Please
insert again:
true
Display message:
The details has been updated
successfully!

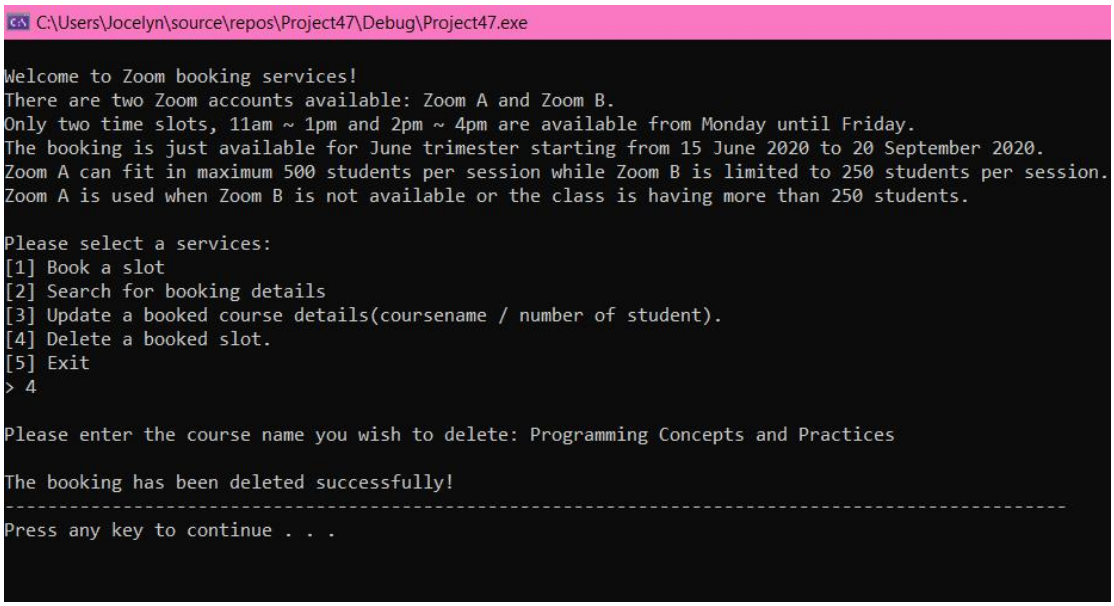

If
select==4
true
Display message:
Please enter the coursename
you wish to delete:
false
Input
coursename
If inFile
is open
false
Display message:
Unable to open file!
true
Display message:
Deleted Detail No:
Zoom:
Slot:
Coursename:
Number of student:
Date:
Display message:
The booking has
been deleted
successfully!
If
select==5
true
Display message:
Thanks!
return 0

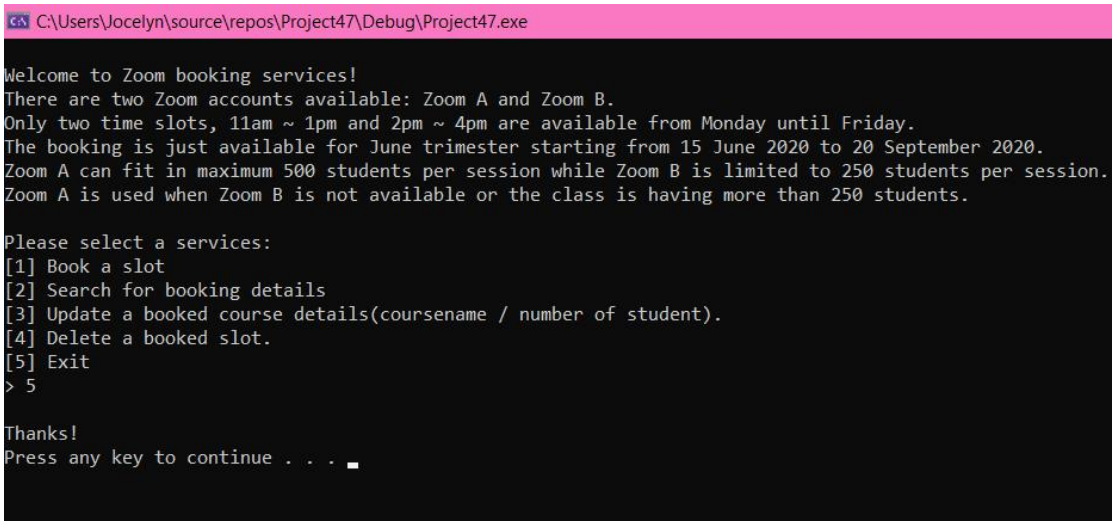
Test cases


C:\Users\Jocelyn\source\repos\Project47\Debug\Project47.exe
Welcome to Zoom booking services!
There are two Zoom accounts available: Zoom A and Zoom B.
Only two time slots, 11am ~ 1pm and 2pm ~ 4pm are available from Monday until Friday.
The booking is just available for June trimester starting from 15 June 2020 to 20 September 2020.
Zoom A can fit in maximum 500 students per session while Zoom B is limited to 250 students per session.
Zoom A is used when Zoom B is not available or the class is having more than 250 students.
Please select a services:
[1] Book a slot
[2] Search for booking details
[3] Update a booked course details(coursename / number of student).
[4] Delete a booked slot.
[5] Exit
> 1
Insert course name: Programming Concepts and Practices
Insert number of students: 699
The number of students are out of range!
Please insert again: 245
Insert a date (DDMMYYYY) > (15062020)
(DD = Day, MM = Month, YYYY = Year)
> 12052020
The date is out of range!
Please insert again: 17092020
(Total number of students less or equal to 250)
Select a Zoom slot:
[1] Zoom A
[2] Zoom B
> 1
Select a time slot:
[1] 11am-1pm
[2] 2pm-4pm
> 2
Book successfully!
Press any key to continue . . .


C:\Users\Jocelyn\source\repos\Project47\Debug\Project47.exe
Welcome to Zoom booking services!
There are two Zoom accounts available: Zoom A and Zoom B.
Only two time slots, 11am ~ 1pm and 2pm ~ 4pm are available from Monday until Friday.
The booking is just available for June trimester starting from 15 June 2020 to 20 September 2020.
Zoom A can fit in maximum 500 students per session while Zoom B is limited to 250 students per session.
Zoom A is used when Zoom B is not available or the class is having more than 250 students.
Please select a services:
[1] Book a slot
[2] Search for booking details
[3] Update a booked course details(coursename / number of student).
[4] Delete a booked slot.
[5] Exit
> 2
Search by:
[1] Zoom slot
[2] Time slot
[3] Course name
[4] Date
> 4
Insert a date (DDMMYYYY) > (15062020)
(DD = Day, MM = Month, YYYY = Year)
> 06092020
It is weekend!
Please insert again: 09092020
Reserved Detail No. 1
Zoom : Zoom A
Slot : 2pm-4pm
Coursename : DWDDSX
Number of Student : 222
Date : 9092020
Above are the booked details!
Press any key to continue . . .


C:\Users\Jocelyn\source\repos\Project47\Debug\Project47.exe
Welcome to Zoom booking services!
There are two Zoom accounts available: Zoom A and Zoom B.
Only two time slots, 11am ~ 1pm and 2pm ~ 4pm are available from Monday until Friday.
The booking is just available for June trimester starting from 15 June 2020 to 20 September 2020.
Zoom A can fit in maximum 500 students per session while Zoom B is limited to 250 students per session.
Zoom A is used when Zoom B is not available or the class is having more than 250 students.
Please select a services:
[1] Book a slot
[2] Search for booking details
[3] Update a booked course details(coursename / number of student).
[4] Delete a booked slot.
[5] Exit
> 3
Please insert the details of a slot you wish to update.
Insert a date (DDMMYYYY) > (15062020)
(DD = Day, MM = Month, YYYY = Year)
> 15092020
Select a Zoom slot:
[1] Zoom A
[2] Zoom B
> 2
Select a time slot:
[1] 11am-1pm
[2] 2pm-4pm
> 1
There is no booked slot.
There is no booked slot.
Press any key to continue . . .


C:\Users\Jocelyn\source\repos\Project47\Debug\Project47.exe
Welcome to Zoom booking services!
There are two Zoom accounts available: Zoom A and Zoom B.
Only two time slots, 11am ~ 1pm and 2pm ~ 4pm are available from Monday until Friday.
The booking is just available for June trimester starting from 15 June 2020 to 20 September 2020.
Zoom A can fit in maximum 500 students per session while Zoom B is limited to 250 students per session.
Zoom A is used when Zoom B is not available or the class is having more than 250 students.
Please select a services:
[1] Book a slot
[2] Search for booking details
[3] Update a booked course details(coursename / number of student).
[4] Delete a booked slot.
[5] Exit
> 4
Please enter the course name you wish to delete: Programming Concepts and Practices
The booking has been deleted successfully!
Press any key to continue . . .


C:\Users\Jocelyn\source\repos\Project47\Debug\Project47.exe
Welcome to Zoom booking services!
There are two Zoom accounts available: Zoom A and Zoom B.
Only two time slots, 11am ~ 1pm and 2pm ~ 4pm are available from Monday until Friday.
The booking is just available for June trimester starting from 15 June 2020 to 20 September 2020.
Zoom A can fit in maximum 500 students per session while Zoom B is limited to 250 students per session.
Zoom A is used when Zoom B is not available or the class is having more than 250 students.
Please select a services:
[1] Book a slot
[2] Search for booking details
[3] Update a booked course details(coursename / number of student).
[4] Delete a booked slot.
[5] Exit
> 5
Thanks!
Press any key to continue . . .

Source code

```
#include<iostream>
#include<iomanip>
#include<string>
#include<fstream>
#include<cctype>
using namespace std;

//Add
void Add(string, string, char*, int, int);
const int size_1 = 60;
char courseName[size_1], bcourseName[size_1];
int numStudent, bnumStudent;
int date, bdate;
string zoom, bzoom;
string slot, bslot;
bool reserved = false;
ofstream outFile;

//Date Validation
void dateValidation(int&);
int juneWeekend[4] = { 20,21,27,29 }; // 15-30
int julyWeekend[8] = { 4,5,11,12,18,19,25,26 }; // 1-31
int augWeekend[10] = { 1,2,8,9,15,16,22,23,29,30 }; // 1-30
int septWeekend[6] = { 5,6,12,13,19,20 }; // 1-20
int day;
bool weekend, year, outOfRange;

//Search
void Search();
ifstream inFile;

//Store
void Store(string, string, char*, int, int);

//Check
void Check(bool&, string, string, char*, int, int);
void Reserved(bool&, string, int);

//Update
void updateDetails(int, string, string);
void update(string, string, char*, int, int);

//Delete
void deleteSlot();
ofstream temp;

//Global Var
bool valid = false;
```

```
int t;
int n;
int select;

struct menu {
	string line1, line2, line3, line4, line5, line6, line7;
};
menu menuStart = { "","\nWelcome to Zoom booking services!\n" ,"There are two
Zoom accounts available: Zoom A and Zoom B.\n" ,
"Only two time slots, 11am ~ 1pm and 2pm ~ 4pm are available from Monday until
Friday.\n" ,
"The booking is just available for June trimester starting from 15 June 2020 to 20
September 2020.\n"
"Zoom A can fit in maximum 500 students per session while Zoom B is limited to
250 students per session.\n" ,
"Zoom A is used when Zoom B is not available or the class is having more than 250
students.\n" };

int main(void) {
	do {
		n = 0;
		reserved = false;
		valid = false;
		cout << menuStart.line1 << menuStart.line2 << menuStart.line3
			<< menuStart.line4 << menuStart.line5 << menuStart.line6 <<
menuStart.line7;
		cout << "\nPlease select a services:\n[1] Book a slot\n[2] Search for booking
details\n" <<
			"[3] Update a booked course details(coursename / number of
student).\n[4] Delete a booked slot.\n[5] Exit\n> ";
		do {
			if (cin >> select)
				if (select > 0 && select < 6)
					valid = true;
				else
					cout << "\nPlease select from above list: ";
			else {
				cin.clear();
				cin.ignore(10000, '\n');
				cout << "\nIt is not an integer!\nPlease select again: ";
			}
		} while (!valid);
		valid = false;
		cin.ignore();

		if (select == 1) {
			Add(zoom, slot, courseName, numStudent, date);
			system("pause");
			system("CLS");
		}
```

```
        else if (select == 2) {
            Search();
            system("pause");
            system("CLS");
        }
        else if (select == 3) {
            // enter date, slot and zoom;
            cout << "\nPlease insert the details of a slot you wish to update." <<
endl;
            dateValidation(date);
            cout << "\nSelect a Zoom slot:\n[1] Zoom A\n[2] Zoom B\n> ";
            do {
                if (cin >> select)
                    if (select > 0 && select < 3)
                        valid = true;
                    else
                        cout << "\nPlease select from above list: ";
                else {
                    cin.clear();
                    cin.ignore(10000, '\n');
                    cout << "\nIt is not an integer!\nPlease select again: ";
                }
            } while (!valid);
            valid = false;
            cin.ignore();
            if (select == 1)
                zoom = "Zoom A";
            else
                zoom = "Zoom B";
            cout << "\nSelect a time slot:\n[1] 11am-1pm\n[2] 2pm-4pm\n> ";
            do {
                if (cin >> select)
                    if (select > 0 && select < 3)
                        valid = true;
                    else
                        cout << "\nPlease select from above list: ";
                else {
                    cin.clear();
                    cin.ignore(10000, '\n');
                    cout << "\nIt is not an integer!\nPlease select again: ";
                }
            } while (!valid);
            valid = false;
            cin.ignore();
            if (select == 1)
                slot = "11am-1pm";
            else
                slot = "2pm-4pm";
            updateDetails(date, zoom, slot);
```

```
        }
        else if (select == 4) {
            deleteSlot();
        }

    } while (select != 5);
    cout << "\nThanks!" << endl;

    std::system("pause");
    return 0;
}

//=================================================================
==================================
// Function Add;
void Add(string zoom, string slot, char courseName[size_1], int numStudent, int date)
{
    cout << "\nInsert course name: ";
    cin.getline(courseName, size_1);
    for (t = 0; t < size_1; t++) {
        courseName[t] = toupper(courseName[t]);
    }
    cout << "\nInsert number of students: ";
    do {
        if (cin >> numStudent)
            if (numStudent > 0 && numStudent < 501)
                valid = true;
            else
                cout << "\nThe number of students are out of range!\nPlease insert
again: ";
        else {
            cin.clear();
            cin.ignore(10000, '\n');
            cout << "\nIt is not an integer!\nPlease insert again: ";
        }
    } while (!valid);
    valid = false;
    cin.ignore();

    dateValidation(date); // Add- dateValidation;

    if (numStudent > 250) {
        cout << "\n(Total number of students more than 250)\nZoom A selected!" <<
endl;
        zoom = "Zoom A";

        cout << "\nSelect a time slot:\n[1] 11am-1pm\n[2] 2pm-4pm\n> ";
        do {
            if (cin >> select)
                if (select > 0 && select < 3)
```

```
                    valid = true;
                else
                    cout << "\nPlease select from above list: ";
            else {
                cin.clear();
                cin.ignore(10000, '\n');
                cout << "\nIt is not an integer!\nPlease select again: ";
            }
        } while (!valid);
        valid = false;
        cin.ignore();

        if (select == 1) {
            slot = "11am-1pm";
            Check(reserved, zoom, slot, courseName, numStudent, date);
            if (!reserved)
                Store(zoom, slot, courseName, numStudent, date);
        }
        else {
            slot = "2pm-4pm";
            Check(reserved, zoom, slot, courseName, numStudent, date);
            if (!reserved)
                Store(zoom, slot, courseName, numStudent, date);
        }
    }
    else {
        // Zoom A is used when Zoom B is under reserved!
        Reserved(reserved, zoom, date);

        if (reserved == true) {
            reserved = false;
            cout << "\n(Total number of students less or equal to 250)\n(Zoom B is
under reserved!)\nZoom A selected!";
            zoom = "Zoom A";

            cout << "\nSelect a time slot:\n[1] 11am-1pm\n[2] 2pm-4pm\n> ";
            do {
                if (cin >> select)
                    if (select > 0 && select < 3)
                        valid = true;
                    else
                        cout << "\nPlease select from above list: ";
                else {
                    cin.clear();
                    cin.ignore(10000, '\n');
                    cout << "\nIt is not an integer!\nPlease select again: ";
                }
            } while (!valid);
            valid = false;
            cin.ignore();
```

```
            if (select == 1) {
                slot = "11am-1pm";
                Check(reserved, zoom, slot, courseName, numStudent, date);
                if (!reserved)
                    Store(zoom, slot, courseName, numStudent, date);
            }
            else {
                slot = "2pm-4pm";
                Check(reserved, zoom, slot, courseName, numStudent, date);
                if (!reserved)
                    Store(zoom, slot, courseName, numStudent, date);
            }
        }
        else {
            cout << "\n(Total number of students less or equal to 250)\nSelect a
Zoom slot:\n[1] Zoom A\n[2] Zoom B\n> ";
            do {
                if (cin >> select)
                    if (select > 0 && select < 3)
                        valid = true;
                    else
                        cout << "\nPlease select from above list: ";
                else {
                    cin.clear();
                    cin.ignore(10000, '\n');
                    cout << "\nIt is not an integer!\nPlease select again: ";
                }
            } while (!valid);
            valid = false;
            cin.ignore();

            if (select == 1) {
                zoom = "Zoom A";

                cout << "\nSelect a time slot:\n[1] 11am-1pm\n[2] 2pm-4pm\n> ";
                do {
                    if (cin >> select)
                        if (select > 0 && select < 3)
                            valid = true;
                        else
                            cout << "\nPlease select from above list: ";
                    else {
                        cin.clear();
                        cin.ignore(10000, '\n');
                        cout << "\nIt is not an integer!\nPlease select again: ";
                    }
                } while (!valid);
                valid = false;
                cin.ignore();
```

```
            if (select == 1) {
                slot = "11am-1pm";
                Check(reserved, zoom, slot, courseName, numStudent, date);
                if (!reserved)
                    Store(zoom, slot, courseName, numStudent, date);
            }
            else {
                slot = "2pm-4pm";
                Check(reserved, zoom, slot, courseName, numStudent, date);
                if (!reserved)
                    Store(zoom, slot, courseName, numStudent, date);
            }
        }
        else {
            zoom = "Zoom B";

            cout << "\nSelect a time slot:\n[1] 11am-1pm\n[2] 2pm-4pm\n> ";
            do {
                if (cin >> select)
                    if (select > 0 && select < 3)
                        valid = true;
                    else
                        cout << "\nPlease select from above list: ";
                else {
                    cin.clear();
                    cin.ignore(10000, '\n');
                    cout << "\nIt is not an integer!\nPlease select again: ";
                }
            } while (!valid);
            valid = false;
            cin.ignore();

            if (select == 1) {
                slot = "11am-1pm";
                Check(reserved, zoom, slot, courseName, numStudent, date);
                if (!reserved)
                    Store(zoom, slot, courseName, numStudent, date);

            }
            else {
                slot = "2pm-4pm";
                Check(reserved, zoom, slot, courseName, numStudent, date);
                if (!reserved)
                    Store(zoom, slot, courseName, numStudent, date);
            }
        }
    }
}
```

```
	cout <<
"-----------------------------------------------------------------------------------------------\n"
;
}
//
// Add- dateValidation;
void dateValidation(int& date) {
	cout << "\nInsert a date ( DDMMYYYY ) > ( 15062020 )\n( DD = Day, MM =
Month, YYYY = Year )\n> ";
	do {
		weekend = false;
		year = false;
		outOfRange = false;

		do {
			if (cin >> date)
				valid = true;
			else {
				cin.clear();
				cin.ignore(10000, '\n');
				cout << "\nIt is not an integer!\nPlease insert again: ";
			}
		} while (!valid);
		valid = false;
		cin.ignore();

		day = date / 1000000;
		if (date >= 10000000) {
			if (date % 10000 != 2020)
				year = true;
			if (date / 10000 % 100 == 6) {
				for (t = 0; t < 4; t++) {
					if (day < 15 || day > 30)
						outOfRange = true;
					if (day == juneWeekend[t])
						weekend = true;
				}
			}
			else if (date / 10000 % 100 == 7) {
				for (t = 0; t < 8; t++) {
					if (day < 1 || day > 31)
						outOfRange = true;
					if (day == julyWeekend[t])
						weekend = true;
				}
			}
			else if (date / 10000 % 100 == 8) {
				for (t = 0; t < 10; t++) {
					if (day < 1 || day > 31)
						outOfRange = true;
```

```
                if (day == augWeekend[t])
                    weekend = true;
            }
        }
        else if (date / 10000 % 100 == 9) {
            for (t = 0; t < 6; t++) {
                if (day < 1 || day > 20)
                    outOfRange = true;
                if (day == septWeekend[t])
                    weekend = true;
            }
        }
        else
            outOfRange = true;
    }
    else {
        if (date % 10000 != 2020)
            year = true;
        if (date / 10000 % 10 == 6) {
            for (t = 0; t < 4; t++) {
                if (day < 15 || day > 30)
                    outOfRange = true;
                if (day == juneWeekend[t])
                    weekend = true;
            }
        }
        else if (date / 10000 % 10 == 7) {
            for (t = 0; t < 8; t++) {
                if (day < 1 || day > 31)
                    outOfRange = true;
                if (day == julyWeekend[t])
                    weekend = true;
            }
        }
        else if (date / 10000 % 10 == 8) {
            for (t = 0; t < 10; t++) {
                if (day < 1 || day > 31)
                    outOfRange = true;
                if (day == augWeekend[t])
                    weekend = true;
            }
        }
        else if (date / 10000 % 10 == 9) {
            for (t = 0; t < 6; t++) {
                if (day < 1 || day > 20)
                    outOfRange = true;
                if (day == septWeekend[t])
                    weekend = true;
            }
        }
```

```
            else
                outOfRange = true;
        }
        if (year == true)
            cout << "\nIt is not 2020!\nPlease insert again: ";
        else if (weekend == true)
            cout << "\nIt is weekend!\nPlease insert again: ";
        else if (outOfRange == true)
            cout << "\nThe date is out of range!\nPlease insert again: ";

    } while ((weekend || year || outOfRange) == true);
}
//
// Add- Check reserved or not for Update/Keep;
void Check(bool& reserved, string zoom, string slot, char courseName[size_1], int
numStudent, int date) {
    inFile.open("booking.txt");
    if (inFile.is_open()) {
        while (!inFile.eof()) {
            getline(inFile, bzoom);
            getline(inFile, bslot);
            inFile.getline(bcourseName, size_1);
            inFile >> bnumStudent;
            inFile >> bdate;
            inFile.ignore();
            if (bzoom == "")
                break;

            if (date == bdate) {
                if (zoom == bzoom || slot == bslot) {
                    reserved = true;
                    cout << "\nReserved Detail    No. " << n + 1 << endl;
                    cout << "Zoom                    : " << bzoom << endl;
                    cout << "Slot                    : " << bslot << endl;
                    cout << "Coursename              : " << bcourseName << endl;
                    cout << "Number of Student    : " << bnumStudent << endl;
                    cout << "Date                    : " << bdate << endl;
                    n++;
                }
            }
        }
        inFile.close();
        if (reserved) {
            cout << "\nThe slot had booked by the above details." << endl;
            cout << "\nPlease select an option to continue:\n[1] Update\n[2] Keep\n>
";
            do {
                if (cin >> select)
                    if (select > 0 && select < 3)
                        valid = true;
```

```
                    else
                        cout << "\nPlease select from above list: ";
                else {
                    cin.clear();
                    cin.ignore(10000, '\n');
                    cout << "\nIt is not an integer!\nPlease select again: ";
                }
            } while (!valid);
            if (select == 1) {
                cout << endl;
                update(zoom, slot, courseName, numStudent, date);
            }
            else {
                cout << "\nKeep successfully!\n";
            }
        }
    }
}
//
// Add- Store the information into booking.txt;
void Store(string zoom, string slot, char courseName[size_1], int numStudent, int date)
{
    outFile.open("booking.txt", ios::app);
    outFile << zoom << endl;
    outFile << slot << endl;
    outFile << courseName << endl;
    outFile << numStudent << endl;
    outFile << date << endl;
    outFile.close();
    cout << "\nBook successfully!\n";
}
//
// Add- Check either Zoom B had be booked or not;
void Reserved(bool& reserved, string zoom, int date) {
    inFile.open("booking.txt");
    if (inFile.is_open()) {
        while (!inFile.eof()) {
            getline(inFile, bzoom);
            getline(inFile, bslot);
            inFile.getline(bcourseName, size_1);
            inFile >> bnumStudent;
            inFile >> bdate;
            inFile.ignore();
            if (bzoom == "")
                break;

            if (date == bdate) {
                if (bzoom == "Zoom B")
                    reserved = true;
            }
```

```
            }
            inFile.close();
        }
}
//=================================================================
=====================================
// Function Search;
void Search() {
    t = 0;
    cout << "\nSearch by:\n[1] Zoom slot\n[2] Time slot\n[3] Course name\n[4]
Date\n> ";
    do {
        if (cin >> select)
            if (select > 0 && select < 5)
                valid = true;
            else
                cout << "\nPlease select from above list: ";
        else {
            cin.clear();
            cin.ignore(10000, '\n');
            cout << "\nIt is not an integer!\nPlease select again: ";
        }
    } while (!valid);
    valid = false;
    cin.ignore();

    if (select == 1) {
        cout << "\nSelect a Zoom slot:\n[1] Zoom A\n[2] Zoom B\n> ";
        do {
            if (cin >> select)
                if (select > 0 && select < 3)
                    valid = true;
                else
                    cout << "\nPlease select from above list: ";
            else {
                cin.clear();
                cin.ignore(10000, '\n');
                cout << "\nIt is not an integer!\nPlease select again: ";
            }
        } while (!valid);
        valid = false;
        cin.ignore();
        cout << endl;

        if (select == 1)
            zoom = "Zoom A";
        else
            zoom = "Zoom B";

        inFile.open("booking.txt");
```

```
        if (inFile.is_open()) {
            while (!inFile.eof()) {
                getline(inFile, bzoom);
                getline(inFile, bslot);
                inFile.getline(bcourseName, size_1);
                inFile >> bnumStudent;
                inFile >> bdate;
                inFile.ignore();
                if (bzoom == "")
                    break;

                if (zoom == bzoom) {
                    cout << "\nReserved Detail    No. " << n + 1 << endl;
                    cout << "Zoom                    : " << bzoom << endl;
                    cout << "Slot                    : " << bslot << endl;
                    cout << "Coursename              : " << bcourseName << endl;
                    cout << "Number of Student    : " << bnumStudent << endl;
                    cout << "Date                    : " << bdate << endl;
                    n++;
                    reserved = true;
                }
            }
            inFile.close();
            if (reserved)
                cout << "Above are the booked details!" << endl;
            else
                cout << "No booked details!" << endl;
            reserved = false;
        }
        else
            cout << "\nUnable to open file!" << endl;
    }
    else if (select == 2) {
        cout << "\nSelect a time slot:\n[1] 11am-1pm\n[2] 2pm-4pm\n> ";
        do {
            if (cin >> select)
                if (select > 0 && select < 3)
                    valid = true;
                else
                    cout << "\nPlease select from above list: ";
            else {
                cin.clear();
                cin.ignore(10000, '\n');
                cout << "\nIt is not an integer!\nPlease select again: ";
            }
        } while (!valid);
        valid = false;
        cin.ignore();
        cout << endl;
```

```
        if (select == 1)
            slot = "11am-1pm";
        else
            slot = "2pm-4pm";

        inFile.open("booking.txt");
        if (inFile.is_open()) {
            while (!inFile.eof()) {
                getline(inFile, bzoom);
                getline(inFile, bslot);
                inFile.getline(bcourseName, size_1);
                inFile >> bnumStudent;
                inFile >> bdate;
                inFile.ignore();
                if (bzoom == "")
                    break;

                if (slot == bslot) {
                    cout << "\nReserved Detail    No. " << n + 1 << endl;
                    cout << "Zoom                    : " << bzoom << endl;
                    cout << "Slot                    : " << bslot << endl;
                    cout << "Coursename              : " << bcourseName << endl;
                    cout << "Number of Student    : " << bnumStudent << endl;
                    cout << "Date                    : " << bdate << endl;
                    n++;
                    reserved = true;
                }
            }
            inFile.close();
            if (reserved)
                cout << "Above are the booked details!" << endl;
            else
                cout << "No booked details!" << endl;
            reserved = false;
        }
        else
            cout << "\nUnable to open file!" << endl;
    }
    else if (select == 3) {
        cout << "\nInsert course name: ";
        cin.getline(courseName, size_1);
        for (t = 0; t < size_1; t++) {
            courseName[t] = toupper(courseName[t]);
        }
        inFile.open("booking.txt");
        if (inFile.is_open()) {
            while (!inFile.eof()) {
                getline(inFile, bzoom);
                getline(inFile, bslot);
                inFile.getline(bcourseName, size_1);
```

```
                inFile >> bnumStudent;
                inFile >> bdate;
                inFile.ignore();
                if (bzoom == "")
                    break;

                if (strcmp(courseName, bcourseName) == 0) {
                    cout << "\nReserved Detail    No. " << n + 1 << endl;
                    cout << "Zoom                    : " << bzoom << endl;
                    cout << "Slot                    : " << bslot << endl;
                    cout << "Coursename              : " << bcourseName << endl;
                    cout << "Number of Student    : " << bnumStudent << endl;
                    cout << "Date                    : " << bdate << endl;
                    n++;
                    reserved = true;
                }
            }
            inFile.close();
            if (reserved)
                cout << "\nAbove are the booked details!" << endl;
            else
                cout << "\nNo booked details!" << endl;
            reserved = false;
        }
        else
            cout << "\nUnable to open file!" << endl;
}
else if (select == 4) {
        dateValidation(date);
        cout << endl;

        inFile.open("booking.txt");
        if (inFile.is_open()) {
            while (!inFile.eof()) {
                getline(inFile, bzoom);
                getline(inFile, bslot);
                inFile.getline(bcourseName, size_1);
                inFile >> bnumStudent;
                inFile >> bdate;
                inFile.ignore();
                if (bzoom == "")
                    break;

                if (date == bdate) {
                    cout << "\nReserved Detail    No. " << n + 1 << endl;
                    cout << "Zoom                    : " << bzoom << endl;
                    cout << "Slot                    : " << bslot << endl;
                    cout << "Coursename              : " << bcourseName << endl;
                    cout << "Number of Student    : " << bnumStudent << endl;
                    cout << "Date                    : " << bdate << endl;
```

```
                    n++;
                    reserved = true;
                }
            }
            inFile.close();
            if (reserved)
                cout << "\nAbove are the booked details!" << endl;
            else
                cout << "\nNo booked details!" << endl;
            reserved = false;
        }
        else
            cout << "\nUnable to open file!" << endl;
    }
    cout <<
"-------------------------------------------------------------------------------------------------\n"
;
}
//=================================================================
====================================
// Function Update;
void updateDetails(int date, string zoom, string slot) {
    inFile.open("booking.txt");

    if (inFile.is_open()) {
        while (!inFile.eof()) {
            getline(inFile, bzoom);
            getline(inFile, bslot);
            inFile.getline(bcourseName, size_1);
            inFile >> bnumStudent;
            inFile >> bdate;
            inFile.ignore();
            if (bzoom == "")
                break;
            if (date == bdate) {
                if (zoom == bzoom && slot == bslot) {
                    cout << "\nReserved Detail    No. " << n + 1 << endl;
                    cout << "Zoom                    : " << bzoom << endl;
                    cout << "Slot                    : " << bslot << endl;
                    cout << "Coursename              : " << bcourseName << endl;
                    cout << "Number of Student    : " << bnumStudent << endl;
                    cout << "Date                    : " << bdate << endl;
                    inFile.close();
                    inFile.open("booking.txt");
                    temp.open("booking_temp.txt");
                    cout << "\nInsert the new course name: ";
                    cin.getline(courseName, size_1);
                    for (t = 0; t < size_1; t++) {
                        courseName[t] = toupper(courseName[t]);
                    }
```

```
                    cout << "\nInsert the new number of students: ";
                    do {
                        if (zoom == "Zoom A") {
                            if (cin >> numStudent)
                                if (numStudent > 0 && numStudent < 501)
                                    valid = true;
                                else
                                    cout << "\nThe number of students are out of
range!\nPlease insert again: ";
                            else {
                                cin.clear();
                                cin.ignore(10000, '\n');
                                cout << "\nIt is not an integer!\nPlease insert again:
";
                            }
                        }
                        else {
                            if (cin >> numStudent)
                                if (numStudent > 0 && numStudent < 250)
                                    valid = true;
                                else
                                    cout << "\nThe number of students are out of
range!\nPlease insert again: ";
                            else {
                                cin.clear();
                                cin.ignore(10000, '\n');
                                cout << "\nIt is not an integer!\nPlease insert again:
";
                            }
                        }
                    } while (!valid);
                    valid = false;
                    cin.ignore();

                    while (!inFile.eof()) {
                        getline(inFile, bzoom);
                        getline(inFile, bslot);
                        inFile.getline(bcourseName, size_1);
                        inFile >> bnumStudent;
                        inFile >> bdate;
                        inFile.ignore();
                        if (bzoom == "")
                            break;

                        if (date != bdate) {
                            if (zoom != bzoom || slot != bslot) {
                                temp << bzoom << endl;
                                temp << bslot << endl;
                                temp << bcourseName << endl;
                                temp << bnumStudent << endl;
```

```
                                        temp << bdate << endl;
                                    }
                                }
                            }
                            temp << zoom << endl;
                            temp << slot << endl;
                            temp << courseName << endl;
                            temp << numStudent << endl;
                            temp << date << endl;
                            cout << "\nThe details has been updated successfully!" << endl;
                            cout <<
"-------------------------------------------------------------------------------------------------\n"
;
                            inFile.close();
                            temp.close();
                            remove("booking.txt");
                            rename("booking_temp.txt", "booking.txt");
                        }
                        else {
                            cout << "\nPlease just insert the details of ONE slot." << endl;
                            inFile.close();
                        }
                    }
                    else
                        cout << "\nThere is no booked slot." << endl;
                }
            }
            else
                cout << "\nUnable to open file!" << endl;
            system("pause");
            system("CLS");

}
//
void update(string zoom, string slot, char courseName[size_1], int numStudent, int
date) {
    inFile.open("booking.txt");
    temp.open("booking_temp.txt");

    if (inFile.is_open()) {
        while (!inFile.eof()) {
            getline(inFile, bzoom);
            getline(inFile, bslot);
            inFile.getline(bcourseName, size_1);
            inFile >> bnumStudent;
            inFile >> bdate;
            inFile.ignore();
            if (bzoom == "")
                break;
            if (date == bdate) {
```

```
                if (zoom == bzoom || slot == bslot) {
                    continue;
                }
            }
            temp << bzoom << endl;
            temp << bslot << endl;
            temp << bcourseName << endl;
            temp << bnumStudent << endl;
            temp << bdate << endl;

        }
        temp << zoom << endl;
        temp << slot << endl;
        temp << courseName << endl;
        temp << numStudent << endl;
        temp << date << endl;
        cout << "\nThe booking has been updated successfully!" << endl;
        inFile.close();
        temp.close();
        remove("booking.txt");
        rename("booking_temp.txt", "booking.txt");
    }
    else
        cout << "\nUnable to open file!" << endl;
}
//================================================================
==================================
// Function Delete;
void deleteSlot() {
    cout << "\nPlease enter the course name you wish to delete: ";
    cin.getline(courseName, size_1);
    for (t = 0; t < size_1; t++) {
        courseName[t] = toupper(courseName[t]);
    }
    inFile.open("booking.txt");
    temp.open("booking_temp.txt");

    if (inFile.is_open()) {
        while (!inFile.eof()) {
            getline(inFile, bzoom);
            getline(inFile, bslot);
            inFile.getline(bcourseName, size_1);
            inFile >> bnumStudent;
            inFile >> bdate;
            inFile.ignore();
            if (bzoom == "")
                break;

            if (strcmp(courseName, bcourseName) != 0) {
                temp << bzoom << endl;
```

```
                temp << bslot << endl;
                temp << bcourseName << endl;
                temp << bnumStudent << endl;
                temp << bdate << endl;
            }
            else {
                cout << "\nDeleted Detail    No. " << n + 1 << endl;
                cout << "Zoom                    : " << bzoom << endl;
                cout << "Slot                   : " << bslot << endl;
                cout << "Coursename             : " << bcourseName << endl;
                cout << "Number of Student    : " << bnumStudent << endl;
                cout << "Date                   : " << bdate << endl;
                n++;
            }
        }
        cout << "\nThe booking has been deleted successfully!" << endl;
        cout <<
"-------------------------------------------------------------------------------------------------\n"
;
        inFile.close();
        temp.close();
        remove("booking.txt");
        rename("booking_temp.txt", "booking.txt");
    }
    else
        cout << "\nUnable to open file!" << endl;
    system("pause");
    system("CLS");
}
//================================================================
====================================
```