

ASSIGNMENT

Course Code	ESC108A
Course Name	Elements of Computer Science and Engineering
Programme	B. Tech. Computer Science and Engineering
Department	Computer Science and Engineering
Faculty	FET

Name of the Student	ALOK KUMAR SINGH
Reg. No	17ETCS002020
Semester/Year	FIRST
Course Leader/s	Ami Rai E

Declaration Sheet			
Student Name	ALOK KUMAR SINGH		
Reg. No	17ETCS002020		
Programme	B. Tech. Computer Science and Engineering	Semester/Year	FIRST
Course Code	ESC108A		
Course Title	Elements of Computer Science and Engineering		
Course Date		to	
Course Leader	Ami Rai E		
<p>Declaration</p> <p>The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.</p>			
Signature of the Student		Date	
Submission date stamp (by Examination & Assessment Section)			
Signature of the Course Leader and date		Signature of the Reviewer and date	

Declaration Sheet	ii
Contents	iii
List of Figures.....	4
Question No. A	5
A1.1 introduction of IOT and M2M	5
A1.2 Features of IoT	5
A1.3 Features of M2M technology	6
A1.4 The stance taken with justification	7
Question No.B1	8
B1.1 Introduction and Problem-solving approach:	8
B1.2 Algorithm:.....	8
B1.3 Implementation.....	9
B1.4 Results and analysis.....	10
B1.5 Concluding remarks.....	10
Question No. B2	11
B2.1 Introduction and Problem-solving approach	11
B2.2 Algorithm.....	11
B2.3 implementation.....	13
B2.4 Results and analysis.....	14
B2.5 Conclusion with justification of the chosen searching algorithm	16
Question No. B3	17
B3.1 Introduction and Problem-solving approach	17
B3.2 Algorithm/Flow chart	17
B3.3 Implementation.....	20
B3.4 Results and analysis.....	21
B3.5 Concluding remarks (Summary, limitations, improvements)	24
Question No. B	25
B4.1 Introduction and Problem-solving approach	25
B4.2 Algorithm/Flow chart	25
B4.3 Implementation.....	27
B4.4 Results and analysis.....	28
B4.5 Concluding remarks (Summary, limitations, improvements)	32

Figure No.	Title of the figure	Pg.No.
Figure 1.1	Code to calculate hypotenuse of a right-angled triangle using user defined function	9
Figure 1.2	Run successful	9
Figure 1.3	Output of above code in fig 1.1	10
Figure 2.1	Code to enter the details of students using structure and searching on the availability of student on the basis of registration number	13
Figure 2.2	Output of above program when student is present	14
Figure 2.3	Output of above program when student is present	14
Figure 2.4	When user try to enter details of less than 10 students	15
Figure 3.1	c program for question B3 (part 1)	16
Figure 3.2	c program for question B3 (part 2)	17
Figure 3.3	Compilation successful	17
Figure 3.4	Output of the question B3 (part 1)	22
Figure 3.5	Output of the question B3 (part 2)	23
Figure 3.6	Output of the question B3 (part 3)	24
Figure 4.1	C program for question B4 (part 1)	27
Figure 4.2	C program for question B4 (part 2)	27
Figure 4.3	Compilation successful	28
Figure 4.4	Output of program in B4, when user enter number of mailing list less than 10	28
Figure 4.5	Output of program in B4, when user enter number of mailing list is at least 10 (part 1)	29
Figure 4.6	Output of program in B4, when user enter number of mailing list is at least 10 (it displaying form the file) (part 2)	30
Figure 4.7	Output of program in B4, when user enter number of mailing list is at least 10 (it displaying form the file) (part 3)	31
Figure 4.8	File "mailing_list" in which the details of the mailing list is stored	31

Solution to Question No. A:

A1.1 introduction of IOT and M2M

IOT is the vast network of millions of devices connected to the internet, like cars, smartphones, jet engines, oil drills tablets and almost everything with a sensor on it. It is the ability to transfer data over a network without requiring human to human or human to human-to-computer interaction. internet of things, can be a person with a heart monitor implant, an automobile that has built-in sensors to alert the driver when tire pressure is low or any other natural or man-made object, which consists any IP address and provided with the ability to transfer data over a network.

M2M means “machine to machine”. It describes the interaction of billions of devices and machines that are connected to the internet and to each other. The devices or machines integrate computing capabilities that enable them to capture data about the world around them are share this with each other connected devices, which relates an intelligent network of things or system. In M2M, the processes that are time consuming, can be automated, leaving people free to get on white more useful activities.

A1.2 Features of IoT

some of the key features of the IOT are:

internet of smart living: remote Control Appliances: Switching on and off remotely appliances to avoid accidents and save energy, Weather: Displays outdoor weather conditions such as humidity, temperature, pressure, wind speed and rain levels with ability to transmit data over long distances, Smart Home Appliances: Refrigerators with LCD screen telling what’s inside, food that’s about to expire, ingredients we need to buy and with all the information available on a Smartphone app.

smart cities: Safety: Digital video monitoring, fire control management, public announcement systems, Transportation: Smart Roads and Intelligent High-ways with warning messages and diversions according to climate conditions and unexpected events like accidents or traffic jams, Waste Management: Detection of rubbish levels in containers to optimize the trash collection routes

smart environment: Air Pollution monitoring: Control of CO2 emissions of factories, pollution emitted by cars and toxic gases generated in farms, Forest Fire Detection: Monitoring of combustion gases and preemptive fire conditions to define alert zones, Weather monitoring: weather conditions monitoring such as humidity, temperature, pressure, wind speed and rain, Earthquake Early Detection,

The Internet of Things concept refers to uniquely identifiable things with their virtual representations in an Internet-like structure and IoT solutions comprising a number of components such as:

- (1) Module for interaction with local IoT devices. This module is responsible for acquisition of observations and their forwarding to remote servers for analysis and permanent storage.
- (2) Module for local analysis and processing of observations acquired by IoT devices.
- (3) Module for interaction with remote IoT devices, directly over the Internet. This module is responsible for acquisition of observations and their forwarding to remote servers for analysis and permanent storage.
- (4) Module for application specific data analysis and processing. This module is running on an application server serving all clients. It is taking requests from mobile and web clients and relevant IoT observations as input, executes appropriate data processing algorithms and generates output in terms of knowledge that is later presented to users.
- (5) User interface (web or mobile): visual representation of measurements in a given context (for example on a map) and interaction with the user, i.e. definition of user queries.

A1.3 Features of M2M technology

There are many features of M2M technology, some of the features are the following:

Cost-efficiency- the browser-based software has low cost service that saves our time by improving our efficiency. By Streamline the logistics of collecting and communicating critical information.

Private-label solution - Offer a remote monitoring service as an added benefit to your customers. It offers private-label solutions for sensor and equipment manufacturers and systems integrators.

Digi device cloud integration- our service integrates with Digi's device cloud. It provides complete end-to-end remote monitoring and control any sensors on any wired or wireless network

Message Delivery for sleeping devices: The M2M System shall be able to manage communication towards a sleeping device

Message transmission scheduling: The M2M System shall be able to manage the scheduling of network access and of messaging. It shall be aware of the scheduling delay tolerance of the M2M Application

M2M Service Capabilities discovery and registration: The M2M System shall support mechanisms to allow M2M Applications to discover M2M Service Capabilities offered to them. Additionally, the M2M Device and M2M Gateway shall support mechanisms to allow the registration of its M2M Service Capabilities to the M2M system.

Continuous connectivity: The M2M System shall support continuous connectivity, for M2M applications requesting the same M2M service on a regular and continuous basis. This continuous connectivity may be de-activated upon request of the Application or by an internal mechanism in the M2M Core.

Radio transmission activity indication and control: The radio transmitting parts of the M2M Device/Gateway should be able to provide (if required by particular applications e.g. eHealth) a real-time indication of radio transmission activity to the application on the M2M Device/Gateway, and may be instructed real-time by the application on the M2M Device/Gateway to suspend/resume the radio transmission activity.

A1.4 The stance taken with justification

The expansion of IP networks around the world has made machine to machine communication quicker and easier while using less power. This network allows the best secure and efficient plus cheapest network for the new business opportunities for consumers and suppliers.

Yes, of course, I agree that “CSE department and IT department are equally benefited from machine to machine and internet of things, respectively”. because it having all these capabilities CSE & IT department can race with the emerging technologies and can make life easier so for now because, as we know that the computer is a device which created revolution in human inventions and it made most of the mutual work a cup of tea.

Since most of the people prefer to do most of their confidential manual works online because of the recallability, speed and less cost, available of rare things and the guaranty of secured file etc. such features, is what is expected by the people, hence we can say hat these features CSE and IR people can advance in the technology, which is essential for the growth of any people or a country.

Solution to Question No. B1:

B1.1 Introduction and Problem-solving approach:

According to the question we have to make a function which take two arguments as base and height of a right-angled triangle and calculate the hypotenuse of the right-angled triangle. And the function should return the value of calculated hypotenuse. Where the values of base and height is going to be entered by user. So, in the function we are going to use Pythagoras theorem which states that in the right-angled triangle, the square of hypotenuse is equal to the sum of square of height and base of that triangle (i.e. $C^2=A^2+B^2$, where c is length of hypotenuse, A is length height and B is length of base).

B1.2 Algorithm:

Algorithm hypo_len

(b,l:integer):integer

Var c:integer;

Begin

 c:=sqrt(b*b+l*l);

 Return c;

End

Function main() : void

Begin

 Var b,l,h: integer;

 Read l,b;

 Call function h=hypo_len(b,l);

 Write h;

End

B1.3 Implementation

```
9      * Author: Alok
10     *
11     * Created on 13 October, 2017, 10:26 PM
12     */
13
14     #include <stdio.h>
15     #include <stdlib.h>
16     #include <math.h>
17     int hypo_len(int, int);
18     int main(int argc, char** argv) {
19         int b, l, h;
20         printf("enter the height and base of triangle: ");
21         scanf("%d %d", &l, &b);
22
23         h=hypo_len(b, l);
24
25         printf("hypotenuse of the triangle is:%d ", h);
26
27         return (EXIT_SUCCESS);
28     }
29
30     int hypo_len(int b, int l){
31
32         int h=sqrt((b*b)+(l*l));
33         return h;
34     }
35
36
```

Fig 1.1 code to calculate hypotenuse of a right-angled triangle using user define function

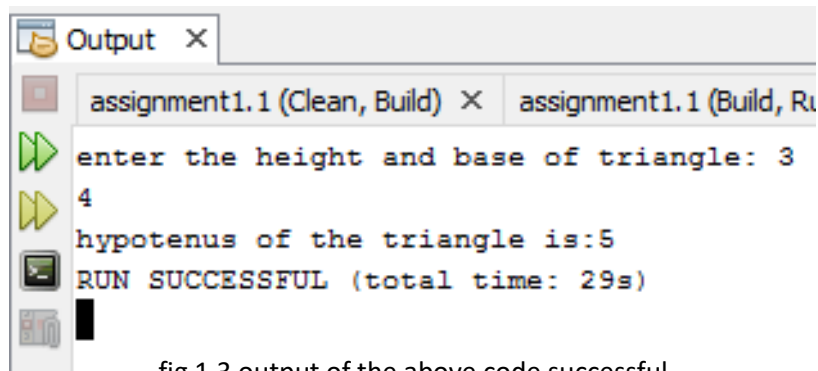
```
make[1]: Leaving directory '/cygdrive/f/net beans/assignment2.1'

CLEAN SUCCESSFUL (total time: 798ms)
cd 'F:\net beans\assignment2.1'
C:\cygwin64\bin\make.exe -f Makefile CONF=Debug
"/usr/bin/make" -f nbproject/Makefile-Debug.mk QMAKE= SUBPROJECTS= .build-conf
make[1]: Entering directory '/cygdrive/f/net beans/assignment2.1'
"/usr/bin/make" -f nbproject/Makefile-Debug.mk dist/Debug/Cygwin-Windows/assignment2.1.exe
make[2]: Entering directory '/cygdrive/f/net beans/assignment2.1'
mkdir -p build/Debug/Cygwin-Windows
rm -f "build/Debug/Cygwin-Windows/main.o.d"
gcc -c -g -MMD -MP -MF "build/Debug/Cygwin-Windows/main.o.d" -o build/Debug/Cygwin-Windows/main.o main.c
mkdir -p dist/Debug/Cygwin-Windows
gcc -o dist/Debug/Cygwin-Windows/assignment2.1 build/Debug/Cygwin-Windows/main.o
make[2]: Leaving directory '/cygdrive/f/net beans/assignment2.1'
make[1]: Leaving directory '/cygdrive/f/net beans/assignment2.1'

BUILD SUCCESSFUL (total time: 1s)
```

fig 1.2 run successful

B1.4 Results and analysis



```
Output x
assignment1.1 (Clean, Build) x assignment1.1 (Build, Run)
>> enter the height and base of triangle: 3
>> 4
>> hypotenous of the triangle is:5
>> RUN SUCCESSFUL (total time: 29s)
```

fig 1.3 output of the above code successful

Here, the user enters 3 and 4 as the length of the 2 sides of the triangle. Then the output (length of the third side must be:

$$\begin{aligned} \text{hypotenous} &= \sqrt{4^2 + 3^2} \\ \text{hypotenous} &= 5 \end{aligned}$$

hence the value of the third side(hypogenous) by the program and the calculated value is same.

B1.5 Concluding remarks

From this program we can easily find the hypotenuse of any right-angled triangle. The user just has to enter the length of height and base, and the program simply output the result as the length of hypotenuse.

The limitation of this program is the size of integer or float i.e. is the user enter the length of base and height beyond the size of its datatype then the output may not be correct.

This program can be improved by adding much field of output, that is, if the user enters any of the three values (value of base, height, hypotenuse). Then the output should be the third side. In this program we are just finding the length of hypotenuse.

Solution to Question No. B2:

B2.1 Introduction and Problem-solving approach

According to the question we have to make a program which inputs the different parameters such as Name, Registration number, Address and Marks of students in a class, from the user. And we have to enter the details of at least 10 students. and after making input from the user we have to find, whether the student is present in the class or not. This whole requirement of the question can be done by using structure. Since a structure is a heterogeneous memory allocation. So, all the parameters we are using can be used inside the structure. And for the multiple entry of details, we are using variable of array type of that structure. And using loop we can enter the multiple details as much as we can. After that, we have to find the on the basis of any parameter that the student is present in the class or not. So, for this I am using registration number to searching the student, and for searching I am using linear search. Because, linear search is very good, if the registration number is available at starting of the variable array of the structure.

B2.2 Algorithm

Structure studenty

Begin

Var reg_no, marks: integer;

Var address [50], name [15] : character;

End s[20];

Algorithm search

(r,n:integer):integer

Var l,flag:=0 : integer

Begin

for l in 0 to n do

begin

if (s[l].reg_no==r)

begin

write "student is present in the class";

flag:=1;

end

end

```

if(flag==0)
begin
write "student is not present in the class";
end
end

algorithm main
begin
var n,l,r : integer;
read m;
if(n<10)
begin
    write "entry of student should not be lee than 10";
    exit(0);
end

for l in 0 to n do
begin
    write "enter the name of student";
    read s[l].name;
    write "enter the registration number of student: ";
    read s[l].reg_no;
    write "enter the address of student: ";
    read s[l].address;
    write "enter the marks of student: ";
    read s[l].marks;
end
write "enter the registration number of the student to be searched";
read r;
search(r,n);
end

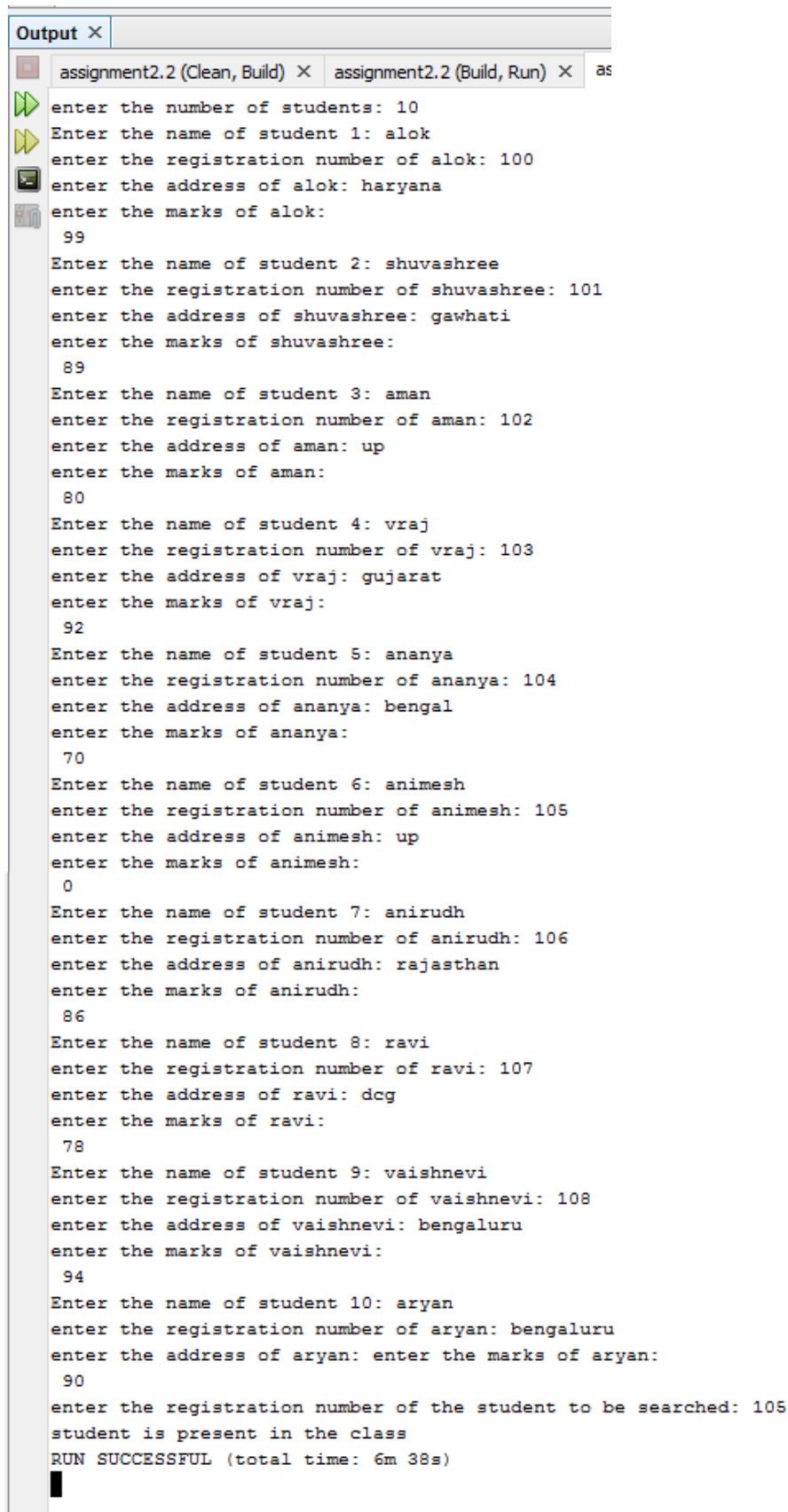
```

B2.3 implementation

```
11  * Author: Alok .....
12  *
13  * Created on 22 October, 2017, 3:21 PM
14  */
15
16  #include <stdlib.h>
17  #include <stdio.h>
18  #include <process.h>
19  int search(int,int);
20
21  struct stydent{
22      int reg_no,marks;
23      char address[50],name[15];
24  }s[20];
25
26
27
28  int main(int argc, char** argv) {
29      int n,i,r;
30      printf("enter the number of students: ");
31      scanf("%d",&n);
32
33      if(n<10){
34          printf("entry of student should not be less than 10");
35          exit(0);
36      }
37      for(i=0;i<n;i++){
38          printf("Enter the name of student %d: ",i+1);
39          scanf("%s",s[i].name);
40          printf("enter the registration number of %s: ",s[i].name);
41          scanf("%d",&s[i].reg_no);
42          printf("enter the address of %s: ",s[i].name);
43          scanf("%s",s[i].address);
44          printf("enter the marks of %s:\n ",s[i].name);
45          scanf("%d",&s[i].marks);
46      }
47      printf("enter the registration number of the student to be searched: ");
48      scanf("%d",&r);
49
50      search(r,n);
51
52      return 0;
53  }
54
55  int search(int r,int n){
56      int i,flag=0;
57
58      for(i=0;i<n;i++){
59          if(s[i].reg_no==r){
60              printf("student is present in the class");
61              flag=1;
62          }
63      }
64
65      if(flag==0)
66          printf("student is not present in the class");
67  }
```

Fig b2.1 code to enter the details of students using structure and searching on the availability of student on basis of registration number.

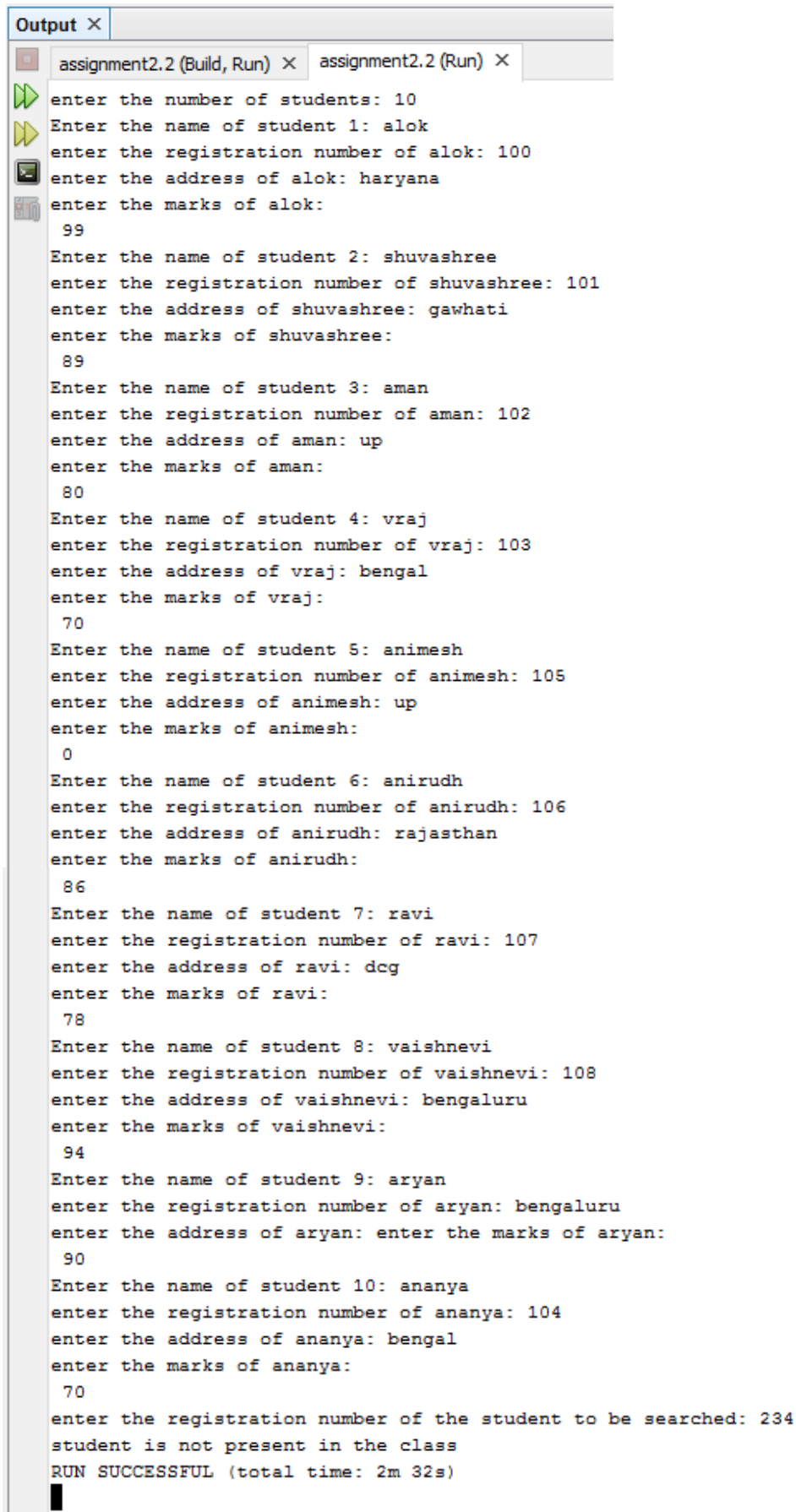
B2.4 Results and analysis



The screenshot shows an IDE's output window with the title 'Output'. It contains the execution results of a program named 'assignment2.2'. The program prompts the user to enter details for 10 students and then searches for a student by registration number. The output shows the following sequence of events:

```
enter the number of students: 10
Enter the name of student 1: alok
enter the registration number of alok: 100
enter the address of alok: haryana
enter the marks of alok:
99
Enter the name of student 2: shuvashree
enter the registration number of shuvashree: 101
enter the address of shuvashree: gawhati
enter the marks of shuvashree:
89
Enter the name of student 3: aman
enter the registration number of aman: 102
enter the address of aman: up
enter the marks of aman:
80
Enter the name of student 4: vraj
enter the registration number of vraj: 103
enter the address of vraj: gujarat
enter the marks of vraj:
92
Enter the name of student 5: ananya
enter the registration number of ananya: 104
enter the address of ananya: bengal
enter the marks of ananya:
70
Enter the name of student 6: animesh
enter the registration number of animesh: 105
enter the address of animesh: up
enter the marks of animesh:
0
Enter the name of student 7: anirudh
enter the registration number of anirudh: 106
enter the address of anirudh: rajasthan
enter the marks of anirudh:
86
Enter the name of student 8: ravi
enter the registration number of ravi: 107
enter the address of ravi: dcg
enter the marks of ravi:
78
Enter the name of student 9: vaishnevi
enter the registration number of vaishnevi: 108
enter the address of vaishnevi: bengaluru
enter the marks of vaishnevi:
94
Enter the name of student 10: aryan
enter the registration number of aryan: bengaluru
enter the address of aryan: enter the marks of aryan:
90
enter the registration number of the student to be searched: 105
student is present in the class
RUN SUCCESSFUL (total time: 6m 38s)
```

Fig b2.2 output of the above program when student is present in the class



```
Output X
assignment2.2 (Build, Run) X assignment2.2 (Run) X
enter the number of students: 10
Enter the name of student 1: alok
enter the registration number of alok: 100
enter the address of alok: haryana
enter the marks of alok:
99
Enter the name of student 2: shuvashree
enter the registration number of shuvashree: 101
enter the address of shuvashree: gawhati
enter the marks of shuvashree:
89
Enter the name of student 3: aman
enter the registration number of aman: 102
enter the address of aman: up
enter the marks of aman:
80
Enter the name of student 4: vraj
enter the registration number of vraj: 103
enter the address of vraj: bengal
enter the marks of vraj:
70
Enter the name of student 5: animesh
enter the registration number of animesh: 105
enter the address of animesh: up
enter the marks of animesh:
0
Enter the name of student 6: anirudh
enter the registration number of anirudh: 106
enter the address of anirudh: rajasthan
enter the marks of anirudh:
86
Enter the name of student 7: ravi
enter the registration number of ravi: 107
enter the address of ravi: dcg
enter the marks of ravi:
78
Enter the name of student 8: vaishnevi
enter the registration number of vaishnevi: 108
enter the address of vaishnevi: bengaluru
enter the marks of vaishnevi:
94
Enter the name of student 9: aryan
enter the registration number of aryan: bengaluru
enter the address of aryan: enter the marks of aryan:
90
Enter the name of student 10: ananya
enter the registration number of ananya: 104
enter the address of ananya: bengal
enter the marks of ananya:
70
enter the registration number of the student to be searched: 234
student is not present in the class
RUN SUCCESSFUL (total time: 2m 32s)
```

Fig b2.2 output of the above program when student is not present in the class

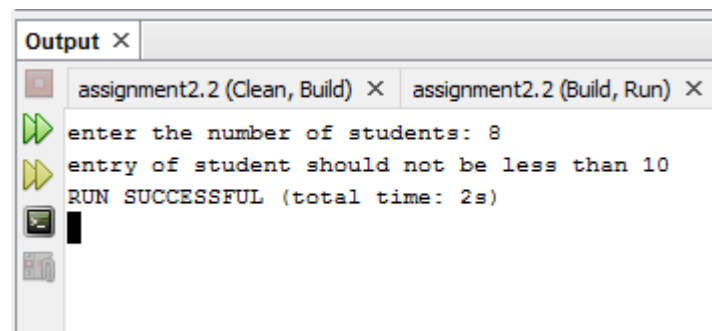


Fig b2.3 when the user tries to enter details of less than 10 students

B2.5 Conclusion with justification of the chosen searching algorithm

From this program a user can store the name, registration number, address and marks of at least 10 students. And after that he/she can search the students for absent or present. This program can be used as record all the details of the students in a class.

this algorithm can be used to beet slow and complexity i.e. this algorithm is good for no complexity and fast response because of the user define function we have used in the above program. To search the availability of student in class, we are using registration number of student because name, address and marks of a student can be same but the registration number of all student I a class is always different. So, for searching, I used linear search. Because in this type of cases the size of array is not much bigger. So, if the registration number is present at the starting of the array, then it will be very good for an algorithm and the time complexity will decrease.

Solution to Question No. B3:

B3.1 Introduction and Problem-solving approach

According to the question, we have to make a structure for mailing list which contain variables for storing name, department, company, city and zip code. the datatype of variable to store name, department, company, city must be in the character type array because these are all set of words. And the datatype of zip code can be integer type because the zip code is always be in the integer value, it can't be float .

From the question we have to make a program in which the user has to enter the mailing list (not less than 10 members). And after that the program should display the mailing list details in either increasing or decreasing order. So, for this we have to use sorting technique to store the details in a particular order.

In the program we must have to use two functions, one function performs the task of displaying the details and the other function perform the task of sorting. As here, I am using bubble sort.

B3.2 Algorithm/Flow chart

struct mailing_list

begin

var zip : integer;

var xompany[20],department[20],name[20],city[20] : character;

end s[20];

Algorithm display (n:integer):voide

Begin

For l in 0 to n do

Begin

Write "name: " s[i].name "department: " s[i].department "city: " s[i].city "company:" s[i].company

"zip code: " s[i].zip;

End

End

Algorithm sort (n:integer) : void

Begin

Var l,j,temp : integer;

Var company[20],department[20],name[20],city[20] : character;

```

For l in 0 to n-1 do
Begin
    For j in 0 to n-1-l do
    Begin
        If(s[j].zip>s[j+1].zip)
        Begin
            temp=s[j].zip;
            s[j].zip=s[j+1].zip;
            s[j+1].zip=temp;

            strcpy(company,s[j].company);
            strcpy(s[j].company,s[j+1].company);
            strcpy(s[j+1].company,company);

            strcpy(department,s[j].department);
            strcpy(s[j].department,s[j+1].department);
            strcpy(s[j+1].department,department);

            strcpy(city,s[j].city);
            strcpy(s[j].city,s[j+1].city);
            strcpy(s[j+1].city,city);

            strcpy(name,s[j].name);
            strcpy(s[j].name,s[j+1].name);
            strcpy(s[j+1].name,name);
        end
    end
end
write "\n displaying the mailing list after shorting: \n";
display(n);
end

algorithm main () :void

```

```
begin
var l,n : integer;
read n;

for l in 0 to n do
begin
    write "enter the name";
    read s[l].name;
    write "enter the department";
    read s[l].department;
    write "enter the company name: ";
    read s[l].company;
    write "enter the name of city: ";
    read s[l].city
    write "enter the zip code: "
    read s[l].zip;
end
write "details of the students before sorting: ";
display(n);
sort(n);

return (0);
end
```

B3.3 Implementation

```
9      hor: Alok
10
11      ated on 28 October, 2017, 8:54 PM
12
13
14      de <stdlib.h>
15      de<stdio.h>
16      de<string.h>
17      de<process.h>
18      isplay(int);
19      ort(int);
20
21      mailing_list{
22      t zip;
23      ar company[20],department[20],name[20],city[20];
24      ;
25
26      in(int argc, char** argv) {
27      t i,n;
28      intf("enter the number of mail list: ");
29      anf("%d",&n);
30
31      r(i=0;i<n;i++){
32      printf("Enter the name: ");
33      scanf("%s",&s[i].name);
34      printf("enter the department of %s: ",s[i].name);
35      scanf("%s",&s[i].department);
36
37      printf("enter the company of %s: ",s[i].name);
38      scanf("%s",&s[i].company);
39      printf("enter the name of city of %s: ",s[i].city);
40      scanf("%s",&s[i].city);
41      printf("enter the zip code on %s: ",s[i].name);
42      scanf("%d",&s[i].zip);
43
44      }
45      printf("\ndetails of the students before sorting: \n");
46      display(n);
47      sort(n);
48
49      return 0;
50
51      }
52
53      void display(int n){
54
55      for(int i=0;i<n;i++){
56      printf("Name: %s \nDepartment: %s\nCompany: %s\ncity: %s\nzip code: %d"
57      \n\n",s[i].name,s[i].department,s[i].company,s[i].city,s[i].zip);
58      printf("\n");
59      }
60
61      }
62
63      void sort(int n){
64      int i,j,temp;
65      char company[20],department[20],name[20],city[20];
66
67      for(i=0;i<n-1;i++){
68      for(j=0;j<n-i-1;j++){
69      if(s[j].zip>s[j+1].zip){
```

Fig 3.1 c program for question B3 (part 1)

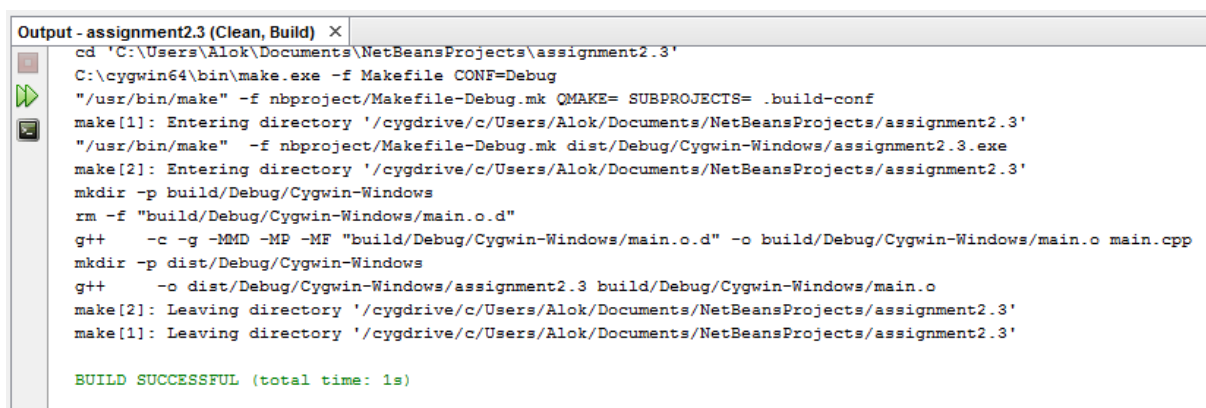
```

65         temp=s[j].zip;
66         s[j].zip=s[j+1].zip;
67         s[j+1].zip=temp;
68
69
70         strcpy(company,s[j].company);
71         strcpy(s[j].company,s[j+1].company);
72         strcpy(s[j+1].company,company);
73
74         strcpy(department,s[j].department);
75         strcpy(s[j].department,s[j+1].department);
76         strcpy(s[j+1].department,department);
77
78         strcpy(city,s[j].city);
79         strcpy(s[j].city,s[j+1].city);
80         strcpy(s[j+1].city,city);
81
82         strcpy(name,s[j].name);
83         strcpy(s[j].name,s[j+1].name);
84         strcpy(s[j+1].name,name);
85     }
86 }
87
88 printf("displaying the mailing list after shorting: \n");
89     display(n);
90 }

```

Fig 3.2 c program for question B3 (part 2)

B3.4 Results and analysis



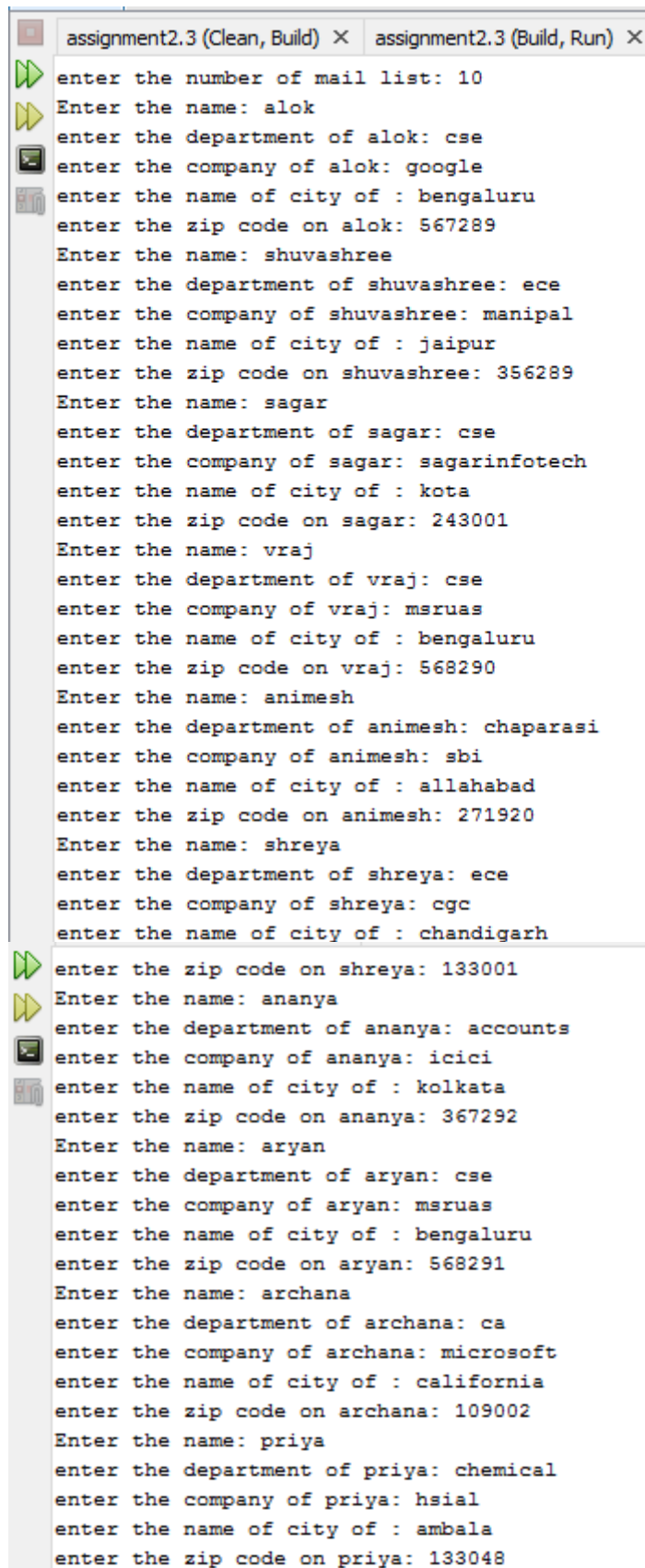
```

Output - assignment2.3 (Clean, Build) X
cd 'C:\Users\Alok\Documents\NetBeansProjects\assignment2.3'
C:\cygwin64\bin\make.exe -f Makefile CONF=Debug
"/usr/bin/make" -f nbproject/Makefile-Debug.mk QMAKE= SUBPROJECTS= .build-conf
make[1]: Entering directory '/cygdrive/c/Users/Alok/Documents/NetBeansProjects/assignment2.3'
"/usr/bin/make" -f nbproject/Makefile-Debug.mk dist/Debug/Cygwin-Windows/assignment2.3.exe
make[2]: Entering directory '/cygdrive/c/Users/Alok/Documents/NetBeansProjects/assignment2.3'
mkdir -p build/Debug/Cygwin-Windows
rm -f "build/Debug/Cygwin-Windows/main.o.d"
g++ -c -g -MMD -MP -MF "build/Debug/Cygwin-Windows/main.o.d" -o build/Debug/Cygwin-Windows/main.o main.cpp
mkdir -p dist/Debug/Cygwin-Windows
g++ -o dist/Debug/Cygwin-Windows/assignment2.3 build/Debug/Cygwin-Windows/main.o
make[2]: Leaving directory '/cygdrive/c/Users/Alok/Documents/NetBeansProjects/assignment2.3'
make[1]: Leaving directory '/cygdrive/c/Users/Alok/Documents/NetBeansProjects/assignment2.3'

BUILD SUCCESSFUL (total time: 1s)

```

Fig 3.3 compilation successful



```
assignment2.3 (Clean, Build) X assignment2.3 (Build, Run) X
>> enter the number of mail list: 10
>> Enter the name: alok
>> enter the department of alok: cse
>> enter the company of alok: google
>> enter the name of city of : bengaluru
>> enter the zip code on alok: 567289
Enter the name: shuvashree
enter the department of shuvashree: ece
enter the company of shuvashree: manipal
enter the name of city of : jaipur
enter the zip code on shuvashree: 356289
Enter the name: sagar
enter the department of sagar: cse
enter the company of sagar: sagarinfotech
enter the name of city of : kota
enter the zip code on sagar: 243001
Enter the name: vraj
enter the department of vraj: cse
enter the company of vraj: msruas
enter the name of city of : bengaluru
enter the zip code on vraj: 568290
Enter the name: animesh
enter the department of animesh: chaparasi
enter the company of animesh: sbi
enter the name of city of : allahabad
enter the zip code on animesh: 271920
Enter the name: shreya
enter the department of shreya: ece
enter the company of shreya: cgc
enter the name of city of : chandigarh
>> enter the zip code on shreya: 133001
>> Enter the name: ananya
>> enter the department of ananya: accounts
>> enter the company of ananya: icici
>> enter the name of city of : kolkata
>> enter the zip code on ananya: 367292
Enter the name: aryan
enter the department of aryan: cse
enter the company of aryan: msruas
enter the name of city of : bengaluru
enter the zip code on aryan: 568291
Enter the name: archana
enter the department of archana: ca
enter the company of archana: microsoft
enter the name of city of : california
enter the zip code on archana: 109002
Enter the name: priya
enter the department of priya: chemical
enter the company of priya: hsial
enter the name of city of : ambala
enter the zip code on priya: 133048
```

Fig 3.4 output of the question B3 (part 1)

```
Name: archana
Department: ca
Company: microsoft
city: california
zip code: 109002

Name: priya
Department: chemical
Company: hsial
city: ambala
zip code: 133048

displaying the mailing list after sorting:

Name: archana
Department: ca
Company: microsoft
city: california
zip code: 109002

Name: shreya
Department: ece
Company: cgc
city: chandigarh
zip code: 133001

Name: priya
Department: chemical
Company: hsial
city: ambala
zip code: 133048

Name: sagar
Department: cse
Company: sagarinfotech
city: kota
zip code: 243001

Name: animesh
Department: chaparasi
Company: sbi
city: allahabad
zip code: 271920

Name: shuvashree
Department: ece
Company: manipal
city: jaipur
zip code: 356289
```

Fig 3.5 output of the question B3 (part 2)

```
Name: ananya
Department: accounts
Company: icici
city: kolkata
zip code: 367292

Name: alok
Department: cse
Company: google
city: bengaluru
zip code: 567289

Name: vraj
Department: cse
Company: msruas
city: bengaluru
zip code: 568290

Name: aryan
Department: cse
Company: msruas
city: bengaluru
zip code: 568291

RUN SUCCESSFUL (total time: 7m 17s)
```

Fig 3.6 output of the question B3 (part 3)

B3.5 Concluding remarks (Summary, limitations, improvements)

The above program will be able to hold the data for a mailing list such as Name, Department, Company, City and Zip Code. and on the basis of zip code it will arrange the list in ascending order.

Limitations of the program is the size of structure variable and the size of the member in the structure. i.e. if the user wants to enter the mailing list of more than 20 then he/she cannot do it, as the size of the structure variable as array is 20.

The improvement of the above program is to apply selection sort instead of bubble sort because there are many parameters to be swapped by the sorting of zip code. so, it will be less in the steps by selection sort instead of bubble sort.

Solution to Question No. B4:**B4.1 Introduction and Problem-solving approach**

According to the question, we have to make a structure for mailing list which contain variables for storing name, department, company, city and zip code. the datatype of variable to store name, department, company, city must be in the character type array because these are all set of words. And the datatype of zip code can be integer type because the zip code is always being in the integer value, it can't be float.

From the question we have to make a program in which the user has to enter the mailing list (not less than 10 members). After that, the mailing list should be stored permanently in the file from the terminal and then we have to display the mailing list from the file to output terminal. So here I am using binary file whose name will be "mailing_list", and inside that file the details will get stored.

B4.2 Algorithm/Flow chart

Algorithm display(var n : integer) : void

Begin

FILE *fp;

Fg=fopen("mailing_list.dat","rb");

If(!fp)

Begin

Write "cant open file";

Exit(0);

End

For I in 0 to n

Begin

Fread(&s2[i],sizeof(s2[i]),1,fp);

Write "department: " s[i].department "company: " s[i].company "city: " s[i].city "zip code: "

s[i].zip;

End

Fclose(fp);

end

algorithm main () : void

begin

```

var l,n : integer;
FILE *fp;
Fp=fopen("mailing_list.dat","wb");
if(!fp)
begin
    printf("cant open file");
    exit(0);
end

write "enter the number of mailing list: ";
read n;
if(n<10)
begin
    write "the entry of the mailing list should not be less than 10";
    exit (0);
end
for l in 0 to n do
begin
    write "enter the name: ";
    read s[l].name;
    write "enter the department: ";
    read s[l].department;
    write "enter the company name: ";
    read s[l].company;
    write "enter the city: ";
    read s[l].city;
    write "enter the zip code: ";
    read s[l].zip;
    fwrite(&s[l], sizeof(s[l]),1,fp);
end
fclose(fp);
display(n);
return 0;
end

```

B4.3 Implementation

```
9      * Author: Alok
10     *
11     * Created on 28 October, 2017, 11:31 PM
12     */
13
14     #include <stdlib.h>
15     #include <stdio.h>
16     #include <string.h>
17     #include <process.h>
18     void display(int);
19
20     struct mailing_list{
21         int zip;
22         char company[20], department[20], name[20], city[20];
23     }s[50], s2[50];
24
25     int main(int argc, char** argv) {
26         int i, n;
27         FILE *fp;
28         fp=fopen("mailing_list.dat", "wb");
29         if(!fp){
30             printf("cant open file");
31             exit(0);
32         }
33
34         printf("enter the number of mail list: ");
35         scanf("%d", &n);
36
37         if(n<10)
38         {
39             printf("the entry for ");
40             exit(0);
41         }
42
43         for(i=0; i<n; i++){
44             printf("Enter the name: ");
45             scanf("%s", &s[i].name);
46             printf("enter the department of %s: ", s[i].name);
47             scanf("%s", &s[i].department);
48             printf("enter the company of %s: ", s[i].name);
49             scanf("%s", &s[i].company);
50             printf("enter the name of city of %s: ", s[i].city);
51             scanf("%s", &s[i].city);
52             printf("enter the zip code on %s: ", s[i].name);
53             scanf("%d", &s[i].zip);
54             fwrite(&s[i], sizeof(s[i]), 1, fp);
55         }
56         fclose(fp);
57         display(n);
58
59         return 0;
60     }
61     void display(int n){
62         FILE *fp;
63         fp=fopen("mailing_list.dat", "rb");
64         if(!fp){
65             printf("cant open file");
66             exit(0);
67         }
68     }
```

Fig 4.1 c-program for question B4 (part 1)

```

69     for(int i=0;i<n;i++){
70         fread(&s2[i], sizeof(s2[i]), 1, fp);
71         printf("Name: %s \nDepartment: %s\nCompany: %s\ncity: %s\nzip code: %d"
72             \n\n",s2[i].name,s2[i].department,s2[i].company,s2[i].city,s2[i].zip);
73         printf("\n");
74     }
75     fclose(fp);
76 }
77

```

Fig 4.2 c-program for question B4 (part 2)

```

Output ×
assignment2.3 (Clean, Build) × assignment2.3 (Build, Run) × assignment2.3 (Run) × assignment2.4 (Build, Run) × assignment2.4 (Run) × assignmen

make[2]: Entering directory '/cygdrive/c/Users/Alok/Documents/NetBeansProjects/assignment2.4'
mkdir -p build/Debug/Cygwin-Windows
rm -f "build/Debug/Cygwin-Windows/main.o.d"
g++ -c -g -MMD -MP -MF "build/Debug/Cygwin-Windows/main.o.d" -o build/Debug/Cygwin-Windows/main.o main.cpp
mkdir -p dist/Debug/Cygwin-Windows
g++ -o dist/Debug/Cygwin-Windows/assignment2.4 build/Debug/Cygwin-Windows/main.o
make[2]: Leaving directory '/cygdrive/c/Users/Alok/Documents/NetBeansProjects/assignment2.4'
make[1]: Leaving directory '/cygdrive/c/Users/Alok/Documents/NetBeansProjects/assignment2.4'

BUILD SUCCESSFUL (total time: 1s)

```

Fig 4.3 compilation successful

B4.4 [Results and analysis](#)

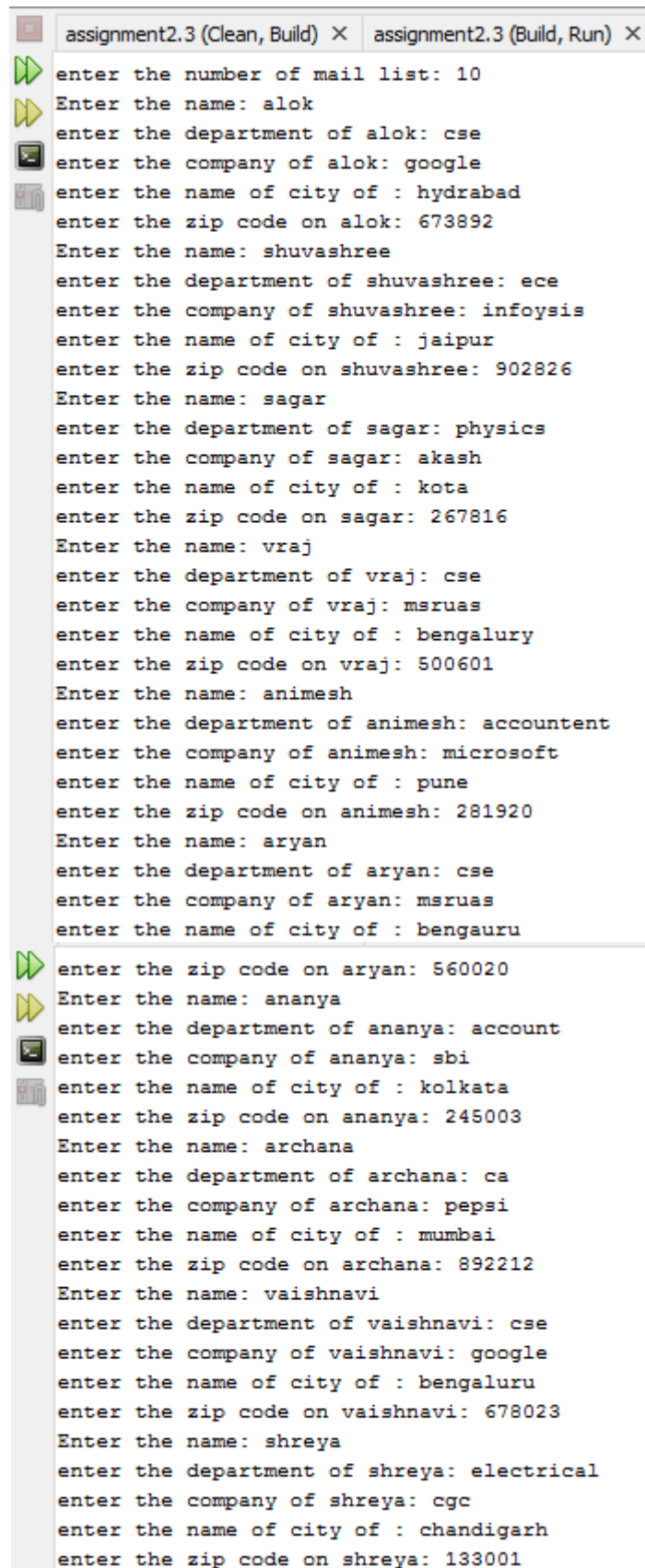
```

Output ×
assignment2.3 (Clean, Build) × assignment2.3 (Build, Run) × assignment2.3 (Run) ×

enter the number of mail list: 5
the entry of the mailing list should not be less than 10
RUN SUCCESSFUL (total time: 1s)

```

Fig 4.4 output of the above program for B4, when user enter number of mailing list is less than 10



```
assignment2.3 (Clean, Build) X assignment2.3 (Build, Run) X
enter the number of mail list: 10
Enter the name: alok
enter the department of alok: cse
enter the company of alok: google
enter the name of city of : hydrabad
enter the zip code on alok: 673892
Enter the name: shuvashree
enter the department of shuvashree: ece
enter the company of shuvashree: infoysis
enter the name of city of : jaipur
enter the zip code on shuvashree: 902826
Enter the name: sagar
enter the department of sagar: physics
enter the company of sagar: akash
enter the name of city of : kota
enter the zip code on sagar: 267816
Enter the name: vraj
enter the department of vraj: cse
enter the company of vraj: msruas
enter the name of city of : bengalury
enter the zip code on vraj: 500601
Enter the name: animesh
enter the department of animesh: accountant
enter the company of animesh: microsoft
enter the name of city of : pune
enter the zip code on animesh: 281920
Enter the name: aryan
enter the department of aryan: cse
enter the company of aryan: msruas
enter the name of city of : bengauru
enter the zip code on aryan: 560020
Enter the name: ananya
enter the department of ananya: account
enter the company of ananya: sbi
enter the name of city of : kolkata
enter the zip code on ananya: 245003
Enter the name: archana
enter the department of archana: ca
enter the company of archana: pepsi
enter the name of city of : mumbai
enter the zip code on archana: 892212
Enter the name: vaishnavi
enter the department of vaishnavi: cse
enter the company of vaishnavi: google
enter the name of city of : bengaluru
enter the zip code on vaishnavi: 678023
Enter the name: shreya
enter the department of shreya: electrical
enter the company of shreya: cgc
enter the name of city of : chandigarh
enter the zip code on shreya: 133001
```

Fig 4.5 output of the above program for B4, when user enter number of mailing list is at least 10 (part 1)

```
Name: alok
Department: cse
Company: google
city: hyderabad
zip code: 673892

Name: shuvashree
Department: ece
Company: infoysis
city: jaipur
zip code: 902826

Name: sagar
Department: physics
Company: akash
city: kota
zip code: 267816

Name: vraj
Department: cse
Company: msruas
city: bengalury
zip code: 500601

Name: animesh
Department: accountent
Company: microsoft
city: pune
zip code: 281920

Name: aryan
Department: cse
Company: msruas
city: bengauru
zip code: 560020

Name: ananya
Department: account
Company: sbi
city: kolkata
zip code: 245003
```

Fig 4.6 output of the above program for B4, when user enter number of mailing list is at least 10 , (it displaying from the file) (part 2)

```

Name: archana
Department: ca
Company: pepsi
city: mumbai
zip code: 892212

Name: vaishnavi
Department: cse
Company: google
city: bengaluru
zip code: 678023

Name: shreya
Department: electrical
Company: cgc
city: chandigarh
zip code: 133001

RUN SUCCESSFUL (total time: 5m 0s)

```

Fig 4.7 output of the above program for B4, when user enter number of mailing list is at least 10 (part 3)

mailing_list - Notepad			
File	Edit	Format	View Help
dH google	cse	alok	hydrabad
æ infoysis	ece	shuvashree	jaipur
(akash	physics	sgara	kota
y msruas	cse	vraj	bengalury
@ microsoft	accountant	animesh	pune
" msruas	cse	aryan	bengaluru
sbi	account	ananya	kolkata
4 pepsi	ca	archana	mumbai
\$X google	cse	vaishnavi	bengaluru
cg	electrical	shreya	chandigarh

Fig 4.8 file “mailing_list” in which the details of the mailing list is stored

B4.5 Concluding remarks (Summary, limitations, improvements)

From the above program we can able to make input of different details in the mailing list and these details will get stored in the file named "maling_list". And from this file, we can access the details again. So here this program is taking input and store that input in the file and the it will show the inputs/details of mailing list from the file.

The limitation of above program is the number of inputs that a user wants make i.e. if the user wants to input details of mailing list less than 10, then he/she cant able to do it. If he/she do so then the program will exit to the terminal by sowing the message than "the entry of the mailing list should not be less than 10". And the other limitations is the size of the variable that is assigned in the structure and structure variable;

The improvement of the above program can be usage of dynamic memory, we can use stacks or queues for the structure variable because if user want to store vary large numbers of mailing list, then he/she cannot able to do so. Because the size of structure variable is pre-defined. If the pre-defined size if very large then the program will take muck memory which will not be in use.

