RAMAIAH UNIVERSITY OF APPLIED SCIENCES

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			

Declaration

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.

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

Contents

Question No. A

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 CO_2 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 <u>The stance taken with justification</u>

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.

Question No.B1

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

```
 * Author: Alok
 *
 * Created on 13 October, 2017, 10:26 PM
 */

#include <stdio.h>
#include <stdlib.h>
#include<math.h>
int hypo_len(int,int);
int main(int argc, char** argv) {
    int b,l,h;
    printf("enter the height and base of triangle: ");
    scanf("%d %d",&l,&b);

    h=hypo_len(b,l);

    printf("hypotenus of the triangle is:%d ",h);

    return (EXIT_SUCCESS);
}

int hypo_len(int b,int l){

    int h=sqrt((b*b)+(l*l));
    return h;
}
```

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

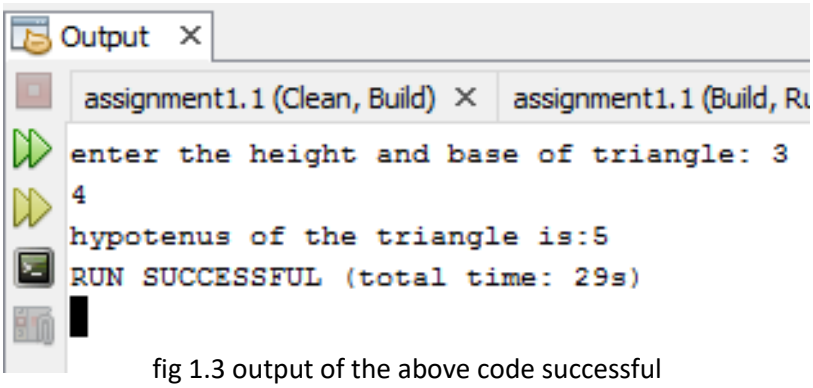

fig 1.3 output of the above code successful

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:

$$hypotenous = \sqrt{4^2 + 3^2}$$
$$hypotenous = 5$$

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.

Question No. B2

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 heterogenous 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 perimeter 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 studeny
Begin
Var reg_no, marks: integer;
Var address [50], name [15] : character;
End s[20];

Algorithm search
        (r,n:integer):integer
Var I,flag:=0 : integer
Begin
for I in 0 to n do
begin
        if (s[i].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,I,r : integer;
read m;
if(n<10)
begin
        write “entry of student should not be lee than 10”;
        exit(0);
end

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

B2.3 implementation

```
 * Author: Alok
 *
 * Created on 22 October, 2017, 3:21 PM
 */

#include <cstdlib>
#include<stdio.h>
#include<process.h>
int search(int,int);

struct stydent{
    int reg_no,marks;
    char address[50],name[15];

}s[20];


int main(int argc, char** argv) {
    int n,i,r;
    printf("enter the number of students: ");
    scanf("%d",&n);

    if(n<10){
        printf("entry of student should not be less than 10");
        exit(0);
    }
    for(i=0;i<n;i++){
        printf("Enter the name of student %d: ",i+1);
      scanf("%s",s[i].name);
        printf("enter the registration number of %s: ",s[i].name);
        scanf("%d",&s[i].reg_no);
        printf("enter the address of %s: ",s[i].name);
        scanf("%s",s[i].address);
        printf("enter the marks of %s:\n ",s[i].name);
        scanf("%d",&s[i].marks);
    }
    printf("enter the registration number of the student to be searched: ");
    scanf("%d",&r);

    search(r,n);

    return 0;
}

int search(int r,int n){
    int i,flag=0;

    for(i=0;i<n;i++){
        if(s[i].reg_no==r){
            printf("student is present in the class");
            flag=1;
        }
    }

    if(flag==0)
        printf("student is not present in the class");
}
```

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

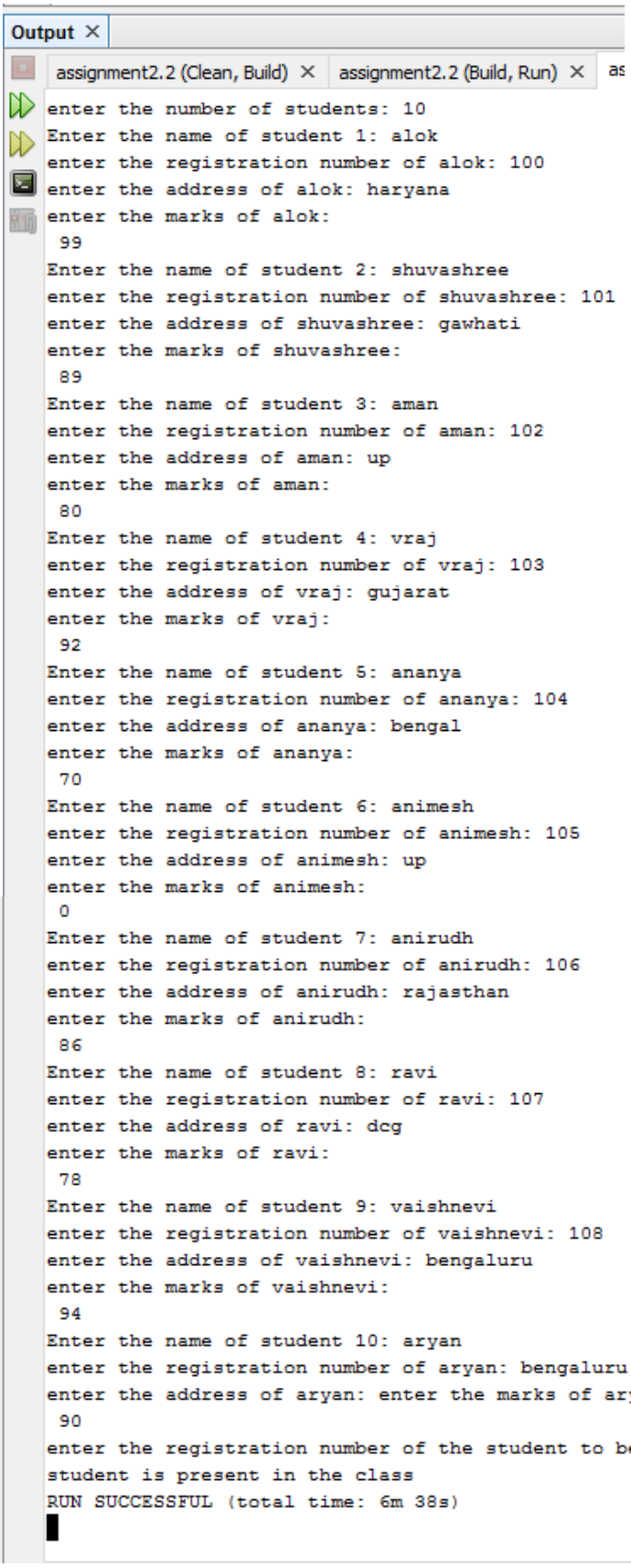

Fig b2.2 output of the above program when student is present in the class

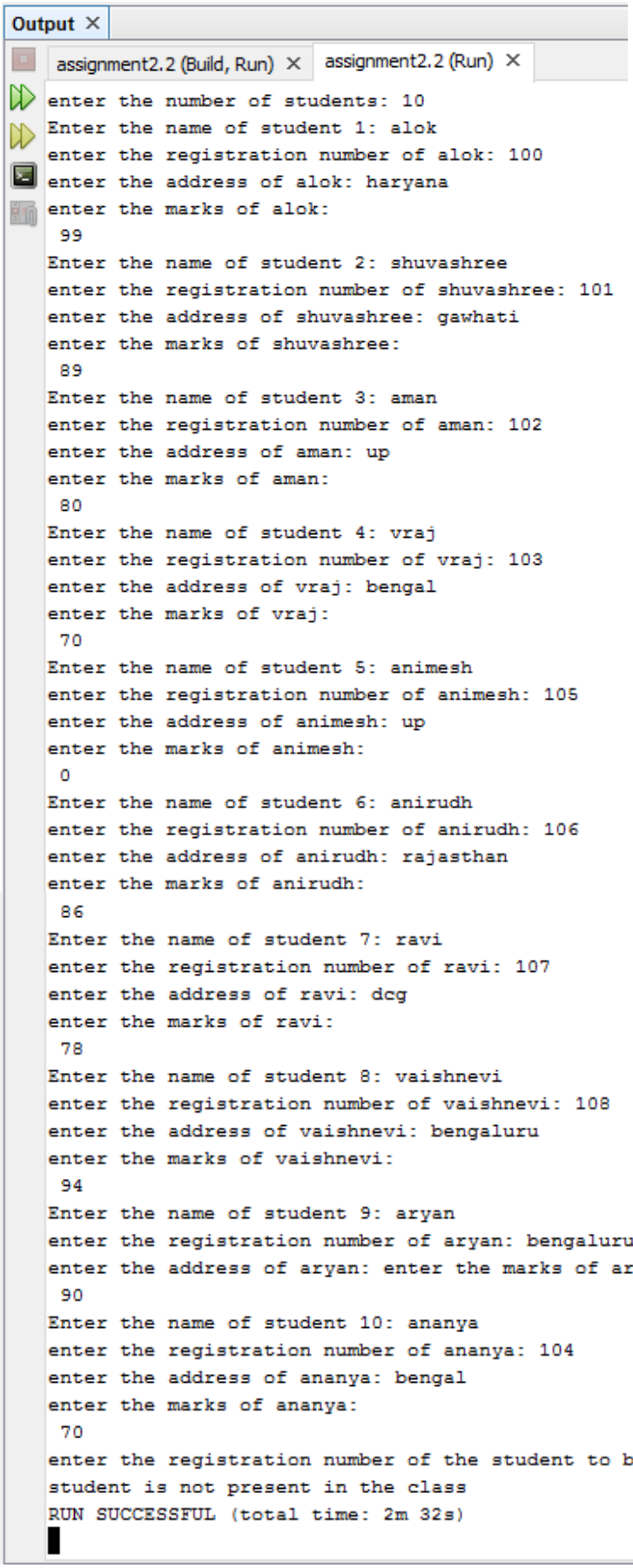

Fig b2.2 output of the above program when student is not present in the class

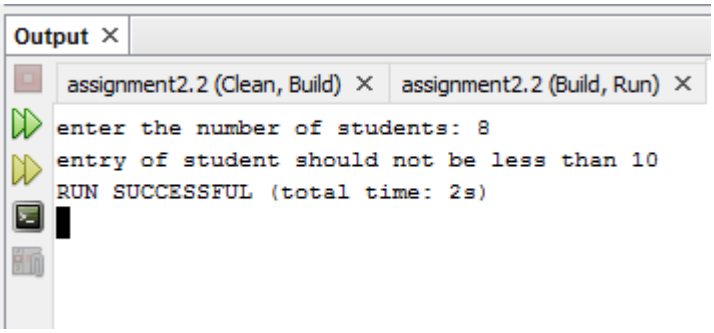

Fig b2.3 when the user tries to enter details of less than 10 students

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.

Question No. B3

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 I 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 I,j,temp : integer;
Var company[20],department[20],name[20],city[20] : character;
```

```
For I in 0 to n-1 do
Begin
        For j in 0 to n-1-I 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 I,n : integer;
read n;

for I in 0 to n do
begin
        write “enter the name”;
        read s[i].name;
        write “enter the department”;
        read s[i].department;
        write “enter the company name: “;
        read s[i].company;
        write “enter the name of city: “;
        read s[i].city
        write “enter the zip code: “
        read s[i].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         printf("enter the company of %s: ",s[i].name);
37         scanf("%s",&s[i].company);
38         printf("enter the name of city of %s: ",s[i].city);
39         scanf("%s",&s[i].city);
40         printf("enter the zip code on %s: ",s[i].name);
41         scanf("%d",&s[i].zip);
42     }
43     printf("\ndetails of the students before sorting: \n");
44     display(n);
45     sort(n);
46
47     return 0;
48  }
49  void display(int n){
50
51     for(int i=0;i<n;i++){
52         printf("Name: %s \nDepartment: %s\nCompany: %s\ncity: %s\nzip code: %d"
53                \n\n",s[i].name,s[i].department,s[i].company,s[i].city,s[i].zip);
           printf("\n");
55     }
56  }
57
58  void sort(int n){
59     int i,j,temp;
60     char company[20],department[20],name[20],city[20];
61
62     for(i=0;i<n-1;i++){
63         for(j=0;j<n-i-1;j++){
64             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)
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

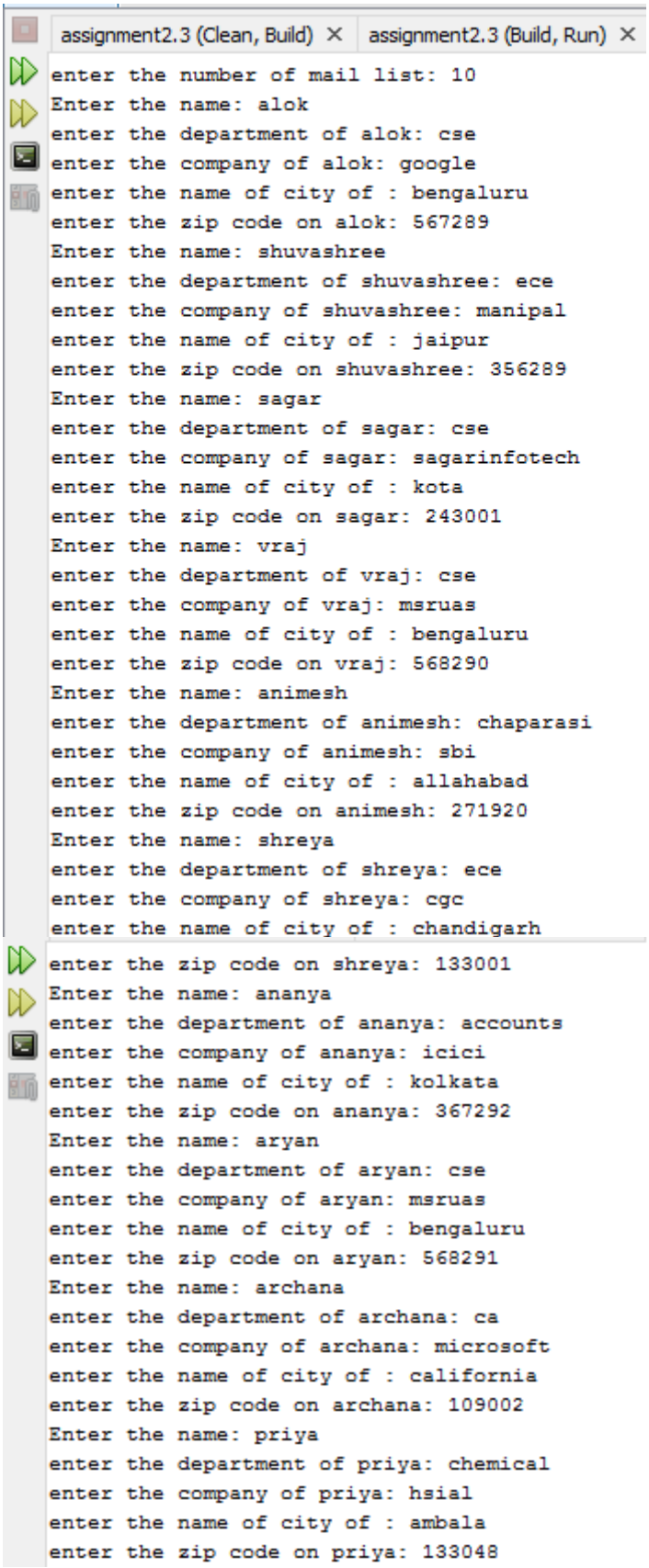

Fig 3.4 output of the question B3 (part 1)

```
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 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 swap by the sorting of zip code. so, it will be less in the steps by selection sort instead of bubble sort.

Question No. B

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 me "maling_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 I,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 I in 0 to n do
        begin
                write “enter the name: “;
                read s[i].name;
                write “enter the department: “;
                read s[i].department;
                write “enter the company name: “;
                read s[i].company;
                write “enter the city: “;
                read s[i].city;
                write “enter the zip code: “;
                read s[i].zip;
                fwrite(&s[i], sizeof(s[i]),1,fp);
        end
fclose(fp);
display(n);
   return 0;
end
```

B4.3 Implementation

```
 * Author: Alok
 *
 * Created on 28 October, 2017, 11:31 PM
 */

#include <cstdlib>
#include<stdio.h>
#include<string.h>
#include<process.h>
void display(int);

struct mailing_list{
    int zip;
    char company[20],department[20],name[20],city[20];
}s[50],s2[50];

int main(int argc, char** argv) {
    int i,n;
    FILE *fp;
    fp=fopen("mailing_list.dat","wb");
    if(!fp){
        printf("cant open file");
        exit(0);
    }

    printf("enter the number of mail list: ");
    scanf("%d",&n);

    if(n<10)
    {
        printf("the entry for ");
        exit(0);
    }

    for(i=0;i<n;i++){
        printf("Enter the name: ");
        scanf("%s",&s[i].name);
        printf("enter the department of %s: ",s[i].name);
        scanf("%s",&s[i].department);
        printf("enter the company of %s: ",s[i].name);
        scanf("%s",&s[i].company);
        printf("enter the name of city of %s: ",s[i].city);
        scanf("%s",&s[i].city);
        printf("enter the zip code on %s: ",s[i].name);
        scanf("%d",&s[i].zip);
        fwrite(&s[i], sizeof(s[i]),1,fp);
    }
    fclose(fp);
    display(n);

    return 0;
}
void display(int n){
    FILE *fp;
    fp=fopen("mailing_list.dat","rb");
      if(!fp){
        printf("cant open file");
        exit(0);
    }
```

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);
            printf("\n");
74      }
75      fclose(fp);
76  }
77
```

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

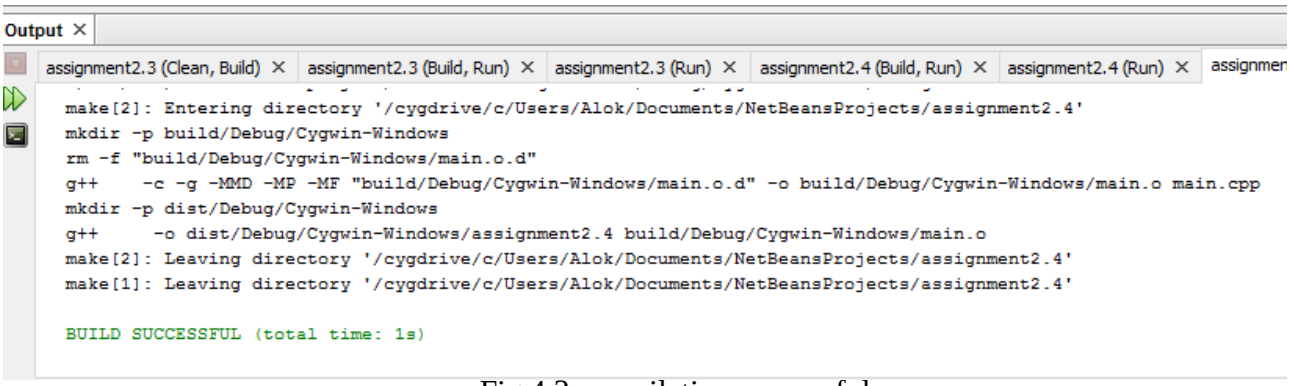

Fig 4.3 compilation successful

B4.4 Results and analysis

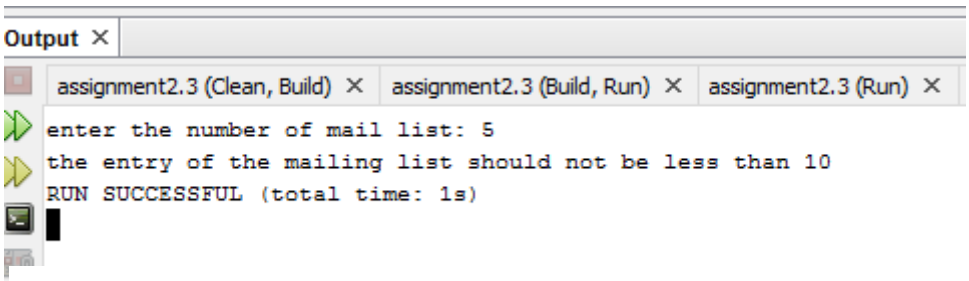

Fig 4.4 output of the above program for B4, when user enter number of mailing list is less than 10

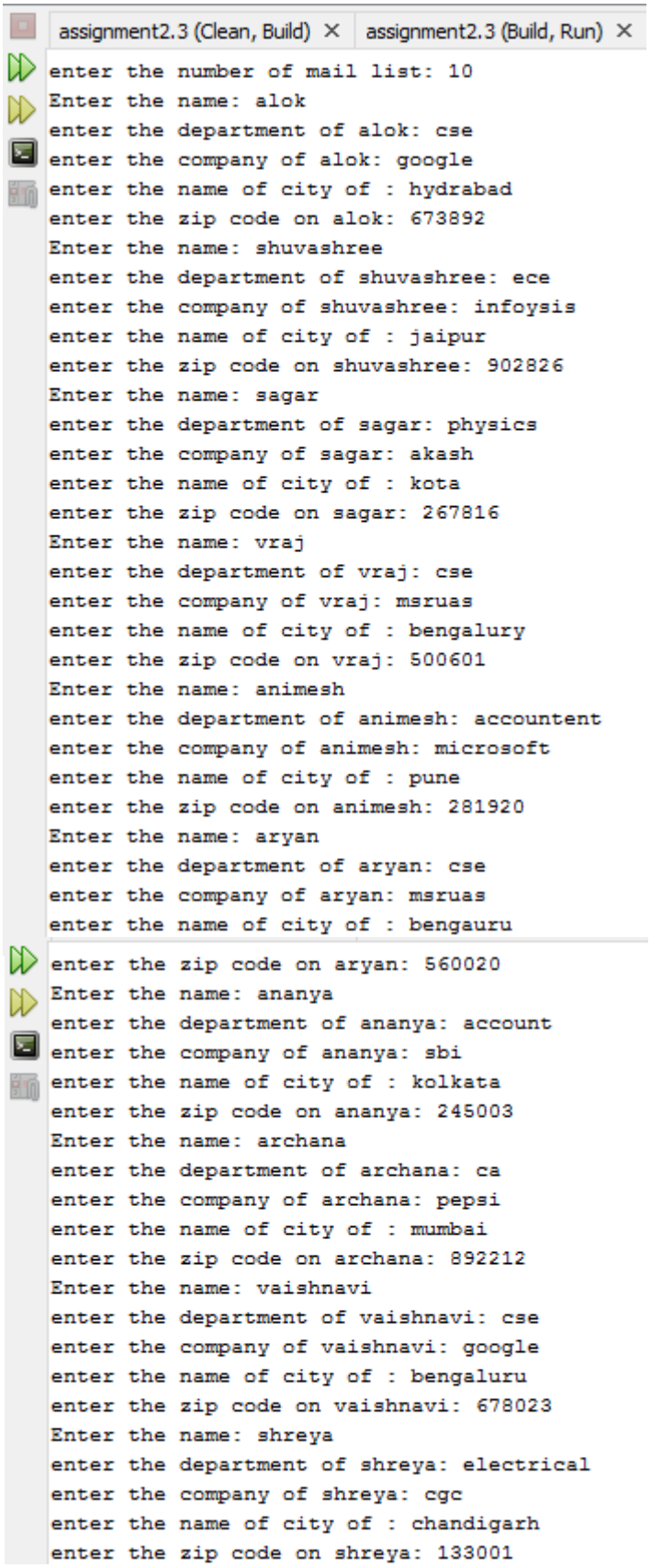

Fig 4.5 output of the above program for B4, when user enter number of mailing list is at least 10 (part 1)

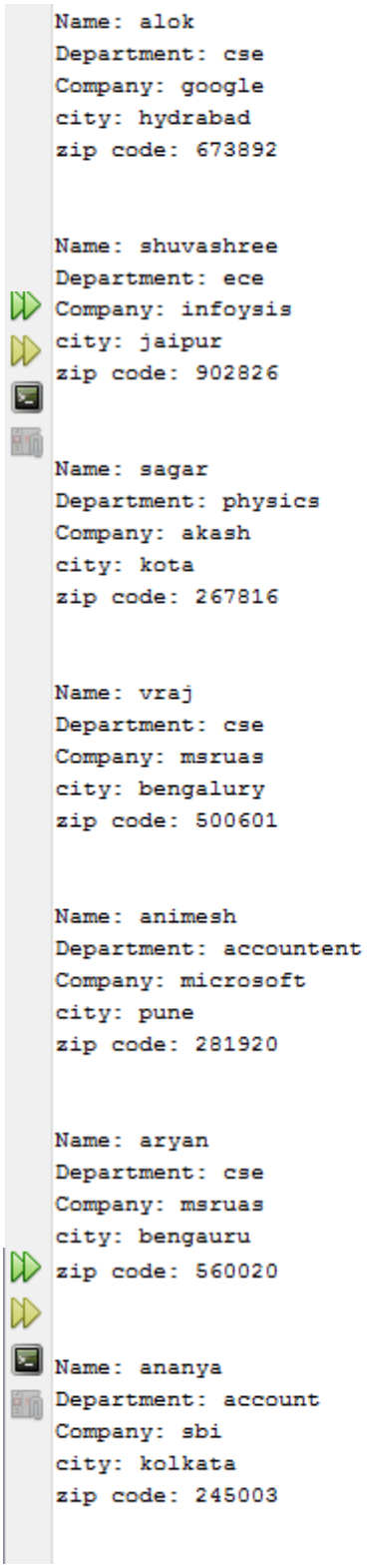

Fig 4.6 output of the above program for B4, when user enter number of mailing list is at least 10 , (it displaying from the file) (part 2)

```
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
ªÆ infoysis	ece	shuvashree	jaipur
(▯ akash	physics	sgara	kota
y£▯ msruas	cse	vraj	bengalury
@M▯ microsoft	accountant	animesh	pune
”‹▯ msruas	cse	aryan	bengaluru
▯¼▯ sbi	account	ananya	kolkata
4 pepsi	ca	archana	mumbai
‡X google	cse	vaishnavi	bengaluru
▯ cgc	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.