

UNIVERSITY OF DODOMA

COLLEGE OF INFORMATICS AND VIRTUAL EDUCATION.



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING.

INDIVIDUAL ASSIGNMENT

NAME: ELIA WILLIAM MARIKI

REGISTRATION NO: T22-03-13063

PROGRAM NAME: SOFTWARE ENGINEERING (SE).

COURSE NAME: INTRODUCTION TO HIGH PROGRAMMING

COURSE CODE: CP 123

### Question:

You have to select a program with a function and loop and tell the kind of problem it was solving and using RAM diagram to show what happening in a computer

### ANSWER:

The following program is used to solve the problem raised during entering the email to a system, this program is used to verify if the email structure or way of writing is valid or invalid.

though doesn't prove that if the email is existing or not existing but only if it is valid or invalid.

### C++ CODE FOR A PROGRAM:

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

bool isValidEmail(string email){

    bool hasAtSymbol = false;
    bool hasDotAfterSymbol = false;

    for(int i=0; i < email.length() ; i++){

        if(email[i]=='@')
        {
            if(hasAtSymbol || i == 0 || i == email.length()-1)
            {
                return false;
            }
            hasAtSymbol = true;
        }
        else if(email[i]=='.')
        {
            if(!hasAtSymbol || hasDotAfterSymbol || i == 0 || i == email.length()-1)
            {
                return false;
            }
            hasDotAfterSymbol = true;
        }

        else if(!isalnum(email[i]) && email[i] != '_' && email[i] != '-')
        {
            return false;
        }

    }

    return hasAtSymbol && hasDotAfterSymbol;

}

int main(){
    string email;
    cout<<"Enter the email Address : ";
    cin>> email;

    if(isValidEmail(email)){
        cout<<"Valid email address"<<endl;
    }
```

```

}
else{
    cout<<"Invalid email address"<<endl;
}
return 0;
}

```

### RAM DIAGRAM FOR ABOVE PROGRAM

This section gives out visual understanding on how RAM in a computer works as the program executes/run

#### Step 1:

variable preparation and declaration - the computer will prepare the variables and even declare them to the reserved space in a RAM. This variable includes

```

bool hasAtSymbol = false;
bool hasDotAfterSymbol = false;
string email;
int i;

```

Here will be three variable 2 of them are boolean and 1 is string type

| RAM      | VARIABLE NAME     |
|----------|-------------------|
| 0        | hasAtSymbol       |
| 0        | hasDotAfterSymbol |
| RESERVED | email             |
| 0        | i                 |

#### Step 2:

The program will prompt the user to enter the Email address

```

Execution of : cout<<"Enter the email Address : ";
               cin>> email;

```

Assume the user enter: [mohm@gmail.com](mailto:mohm@gmail.com)

Then the value will be stored in a RAM as a string

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 0              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| mohm@gmail.com | email             |
| 0              | i                 |

### Step 3:

The program in a condition wise will call a function which returns boolean value as its return type called: `isValidEmail(email)`; and insert the value of email in a function,

Inside the function the program will enter into a loop to with a numbers of conditions to check if the email is valid or not.

The first thing the program will check the length of the email execution of: `email.length()`; where it will find that it is 14

in this loop the condition will be set as the limiting condition for a loop to run/execute shall be less than the `email.length()` which is 14. So here same wise there will be some sub steps in this step.

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 0              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| mohm@gmail.com | email             |
| 0              | i                 |

1<sup>st</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 0              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| mohm@gmail.com | email             |
| 1              | i                 |

2<sup>nd</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 0              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| mohm@gmail.com | email             |
| 2              | i                 |

3<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 0              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| mohm@gmail.com | email             |
| 3              | i                 |

4<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| mohm@gmail.com | email             |
| 4              | i                 |

5<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| mohm@gmail.com | email             |
| 5              | i                 |

6<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| mohm@gmail.com | email             |
| 6              | i                 |

7<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| mohm@gmail.com | email             |
| 7              | i                 |

8<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| mohm@gmail.com | email             |
| 8              | i                 |

9<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| mohm@gmail.com | email             |
| 9              | i                 |

10<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 1              | hasDotAfterSymbol |
| mohm@gmail.com | email             |
| 10             | i                 |

11<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 1              | hasDotAfterSymbol |
| mohm@gmail.com | email             |
| 11             | i                 |

12<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 1              | hasDotAfterSymbol |
| mohm@gmail.com | email             |
| 12             | i                 |

13<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 1              | hasDotAfterSymbol |
| mohm@gmail.com | email             |
| 13             | i                 |

14<sup>th</sup> loop execution

Then the loop will break and from that current function should return a boolean value

execution of : `return hasAtSymbol && hasDotAfterSymbol;`

Here it will return 1 since both has true value to the main function in a that if condition

Hence : it will print out > Valid email address

```

Applications  Places  System
Parrot Terminal
File Edit View Search Terminal Help
[dawillybg@parrot]--[~/Desktop/2nd semester/CP123]
$g++ email\ check.cpp -o email_check
[dawillybg@parrot]--[~/Desktop/2nd semester/CP123]
$./email_check
Enter the email Address : mohm@gmail.com
Valid email address
[dawillybg@parrot]--[~/Desktop/2nd semester/CP123]
$
1 bool hasAtSymbol = false;
2 bool hasDotAfterSymbol = false;
3
4 for(int i=0; i < email.length(); i++){
5     if(email[i] == '@'){
6         hasAtSymbol = true;
7     }
8     if(email[i] == '.'){
9         hasDotAfterSymbol = true;
10    }
11 }
12 if(hasAtSymbol && hasDotAfterSymbol){
13     cout << "Valid email address" << endl;
14 }
15 else{
16     cout << "Invalid email address" << endl;
17 }
18 }
```

FROM:

Step 2: AGAIN

The program will prompt the user to enter the Email address

```
Execution of : cout<<"Enter the email Address : ";  
cin>> email;
```

Assume the user enter: [James@gmailcom](#)

Then the value will be stored in a RAM as a string

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 0              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| James@gmailcom | email             |
| 0              | i                 |

Step 3:AGAIN

The program in a condition wise will call a function which returns boolean value as its return type called: `isValidEmail(email)` ; ,and insert the value of email in a function,

Inside the function the program will enter into a loop to with a numbers of conditions to check if the email is valid or not.

The first thing the program will check the length of the email  
execution of : `email.length()`; where it will find that it is 14

in this loop the condition will be set as the limiting condition for a loop to run/execute shall be less than the `email.length()` which is 14. So here same wise there will be some sub steps in this step.

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 0              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| James@gmailcom | email             |
| 0              | i                 |

1<sup>st</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 0              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| James@gmailcom | email             |
| 1              | i                 |

2<sup>nd</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 0              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| James@gmailcom | email             |
| 2              | i                 |

3<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 0              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| James@gmailcom | email             |
| 3              | i                 |

4<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| James@gmailcom | email             |
| 4              | i                 |

5<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| James@gmailcom | email             |
| 5              | i                 |

6<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| James@gmailcom | email             |
| 6              | i                 |

7<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| James@gmailcom | email             |
| 7              | i                 |

8<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| James@gmailcom | email             |
| 8              | i                 |

9<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| James@gmailcom | email             |
| 9              | i                 |

10<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| James@gmailcom | email             |
| 10             | i                 |

11<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| James@gmailcom | email             |
| 11             | i                 |

12<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| James@gmailcom | email             |
| 12             | i                 |

13<sup>th</sup> loop execution

| RAM            | VARIABLE NAME     |
|----------------|-------------------|
| 1              | hasAtSymbol       |
| 0              | hasDotAfterSymbol |
| James@gmailcom | email             |
| 13             | i                 |

14<sup>th</sup> loop execution

Then the loop will break and from that current function should return a boolean value

execution of : return hasAtSymbol && hasDotAfterSymbol;

Here it will return 0 since one has true value and another false value to the main function in a that if condition

Hence : it will print out > Invalid email address

```
Applications Places System
ParrotTerminal
File Edit View Search Terminal Help
[dawillybg@parrot]~/Desktop/2nd semister/CP123
$ ./email_check
Enter the email Address : James@gmailcom
Invalid email address
[dawillybg@parrot]~/Desktop/2nd semister/CP123
$ cat hashtofilterSymbol.c
#include <stdio.h>
#include <string.h>
#include <ctype.h>
int main()
{
    char email[100];
    printf("Enter email address: ");
    fgets(email, sizeof(email), stdin);
    email[strcspn(email, "\n")] = '\0';
    int i;
    for(i = 0; i < strlen(email); i++)
    {
        if(email[i] != '@' && email[i] != '.' && email[i] != '+' && email[i] != '-' && email[i] != '_' && email[i] != '%' && email[i] != '~' && email[i] != '!' && email[i] != '*' && email[i] != '^' && email[i] != '&' && email[i] != '$' && email[i] != '#' && email[i] != '=' && email[i] != '&' && email[i] != '?' && email[i] != ':' && email[i] != ';' && email[i] != ',' && email[i] != '/' && email[i] != '\\' && email[i] != '`' && email[i] != '~' && email[i] != ' ' && email[i] != '\t' && email[i] != '\n')
        {
            printf("Invalid email address\n");
            return 1;
        }
    }
    printf("Valid email address\n");
    return 0;
}
```

