Name:-SAMHITHA R Roll No:-17ETCS002156

EXPERIMENT 1

Demonstrate the use of data types, local variables and Random function. Tabulate the output for various inputs and verify against expected values. Analyze the efficiency of the algorithm. Describe your learning along with the limitations of overall approach if any. Suggest how these can be overcome in conclusion. Write a C program to illustrate random number generation. Modify the program to generate a random number between lower to upper limit. Your report should include:

1. Introduction and Purpose of Experiment:-
   The purpose of the experiment is to write a program in C language to generate random numbers as specified by the user.
2. Aim and Objectives:-
   To design and develop a C programs using Data types, local variables and Random number generation to demonstrate the use and significate of the same in programming.
3. Design an algorithm for the given problem statement and develop a flowchart/pseudo-code:-

Program for Random number generation without any limits

Pseudo Code:

Algorithm random: int main()

begin

var i;

for i in 1 to 5,step 1 do

begin

write : rand();

end

return 0;

Algorithm:

Step 1:Take a for loop running from the lower limit till the number of random numbers required with an increment of 1.

Step 2:Print the random numbers.

Generating Pseudo random number:

Algorithm random : int main()

srand(time(0));

begin

var i;

for i in 1 to 5,step 1 do

begin

write : rand();

end

return 0;

Program for Random number generation with upper and lower limits.

Algorithm:

Step 1:Input the lower limit

Step 2:Input the upper limit

Step 3:take a for loop running from the lower limit till the number of random numbers required with an increment of 1.

Step 4:Perform the operation that is defined in the pseudo code

Step 5:Print the random numbers.

Pseudo code:

```
Algorithm random: int main()
begin
var Integer: i,n,m,r;
read:lower limit(n);
read:upper limit(m);
for i in m to 10,step 1 do
begin
r = rand() %n+m;
write : r;
end
return 0;
```

4. Implement C program

The function rand() is used to generate random numbers in a program. We can get any number of random numbers by changing the condition in the 'for' loop.

To generate Pseudo random numbers, we need to call a function "srand(time(0))".We need to use a header file <time.h>.

Program for Random number generation without any limits

```c
#include <stdio.h>
#include <stdlib.h>
/*
 * Samhitha R
 * 17ETCS002156
 */
int main(int argc, char** argv) {
    for(int i=0;i<=5;i++)
    {
        printf("%d \n",rand());
    }

    return (EXIT_SUCCESS);
}
```

Fig 1.1.1:Program code 1

```
cd 'C:\Users\DELL\Documents\NetBeansProjects\RandomNumber'
C:\cygwin64\bin\make.exe -f Makefile CONF=Debug clean
"/usr/bin/make" -f nbproject/Makefile-Debug.mk QMAKE= SUBPROJECTS= .clean-conf
make[1]: Entering directory '/cygdrive/c/Users/DELL/Documents/NetBeansProjects/RandomNumber'
rm -f -r build/Debug
make[1]: Leaving directory '/cygdrive/c/Users/DELL/Documents/NetBeansProjects/RandomNumber'

CLEAN SUCCESSFUL (total time: 793ms)
cd 'C:\Users\DELL\Documents\NetBeansProjects\RandomNumber'
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/DELL/Documents/NetBeansProjects/RandomNumber'
"/usr/bin/make"  -f nbproject/Makefile-Debug.mk dist/Debug/Cygwin-Windows/randomnumber.exe
make[2]: Entering directory '/cygdrive/c/Users/DELL/Documents/NetBeansProjects/RandomNumber'
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/randomnumber build/Debug/Cygwin-Windows/main.o
make[2]: Leaving directory '/cygdrive/c/Users/DELL/Documents/NetBeansProjects/RandomNumber'
make[1]: Leaving directory '/cygdrive/c/Users/DELL/Documents/NetBeansProjects/RandomNumber'

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

Fig 1.1.2:Compilation 1

Program for Random number generation with upper and lower limits.

```c
#include <stdio.h>
#include <stdlib.h>
/*SAMHITHA R
 17ETCS002156 */
int main(int argc, char** argv) {
 int i,r,m,n;
 printf("Enter the lower limit");
 scanf("%d",&m);
 printf("Enter the upper limit");
 scanf("%d",&n);
 printf("Random numbers are:\n");
  for(i=m;i<=10;i++)
  {
    r=rand()%n+m;
    printf("%d\n",r);
  }
    return (EXIT_SUCCESS);
}
```

Fig 1.2.1:Program code 2

```
cd 'C:\Users\DELL\Documents\NetBeansProjects\CppApplication_2'
C:\cygwin64\bin\make.exe -f Makefile CONF=Debug clean
"/usr/bin/make" -f nbproject/Makefile-Debug.mk QMAKE= SUBPROJECTS= .clean-conf
make[1]: Entering directory '/cygdrive/c/Users/DELL/Documents/NetBeansProjects/CppApplication_2'
rm -f -r build/Debug
make[1]: Leaving directory '/cygdrive/c/Users/DELL/Documents/NetBeansProjects/CppApplication_2'

CLEAN SUCCESSFUL (total time: 888ms)
cd 'C:\Users\DELL\Documents\NetBeansProjects\CppApplication_2'
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/DELL/Documents/NetBeansProjects/CppApplication_2'
"/usr/bin/make"  -f nbproject/Makefile-Debug.mk dist/Debug/Cygwin-Windows/cppapplication_2.exe
make[2]: Entering directory '/cygdrive/c/Users/DELL/Documents/NetBeansProjects/CppApplication_2'
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/cppapplication_2 build/Debug/Cygwin-Windows/main.o
make[2]: Leaving directory '/cygdrive/c/Users/DELL/Documents/NetBeansProjects/CppApplication_2'
make[1]: Leaving directory '/cygdrive/c/Users/DELL/Documents/NetBeansProjects/CppApplication_2'

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

Fig 1.2.2:Compilation 2

```c
#include <stdio.h>
#include <stdlib.h>
#include<time.h>
/*
 * Samhitha R
 * 17ETCS002156
 */
int main(int argc, char** argv) {
    srand(time(0));
    for(int i=0;i<=5;i++)
    {
        printf("%d \n",rand());
    }

    return (EXIT_SUCCESS);
}
```

Fig 1.3.1:Program code 3

```
cd 'C:\Users\DELL\Documents\NetBeansProjects\RandomNumber'
C:\cygwin64\bin\make.exe -f Makefile CONF=Debug clean
"/usr/bin/make" -f nbproject/Makefile-Debug.mk QMAKE= SUBPROJECTS= .clean-conf
make[1]: Entering directory '/cygdrive/c/Users/DELL/Documents/NetBeansProjects/RandomNumber'
rm -f -r build/Debug
make[1]: Leaving directory '/cygdrive/c/Users/DELL/Documents/NetBeansProjects/RandomNumber'

CLEAN SUCCESSFUL (total time: 836ms)
cd 'C:\Users\DELL\Documents\NetBeansProjects\RandomNumber'
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/DELL/Documents/NetBeansProjects/RandomNumber'
"/usr/bin/make"  -f nbproject/Makefile-Debug.mk dist/Debug/Cygwin-Windows/randomnumber.exe
make[2]: Entering directory '/cygdrive/c/Users/DELL/Documents/NetBeansProjects/RandomNumber'
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/randomnumber build/Debug/Cygwin-Windows/main.o
make[2]: Leaving directory '/cygdrive/c/Users/DELL/Documents/NetBeansProjects/RandomNumber'
make[1]: Leaving directory '/cygdrive/c/Users/DELL/Documents/NetBeansProjects/RandomNumber'

BUILD SUCCESSFUL (total time: 4s)
```

Fig 1.3.2: Compilation 3

5. Presentation of Results

```
1481765933
1085377743
1270216262
1191391529
812669700
553475508

RUN SUCCESSFUL (total time: 2s)
```

6. Fig 1.1.3:Output 1

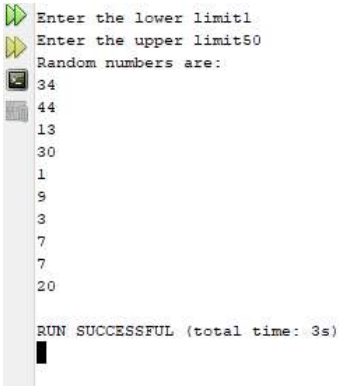

Fig 1.2.3:Output 2

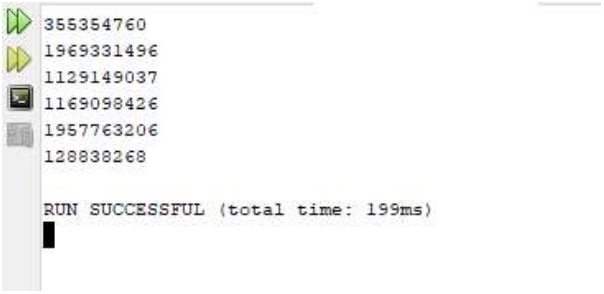

Fig 1.3.3:Output 3

7. Conclusion

Random numbers were generated using the function rand().
In a normal random function ,the same set of random numbers are generated every time the program is executed.
In pseudo random function ,a different set of random numbers are generated when executed every time.