

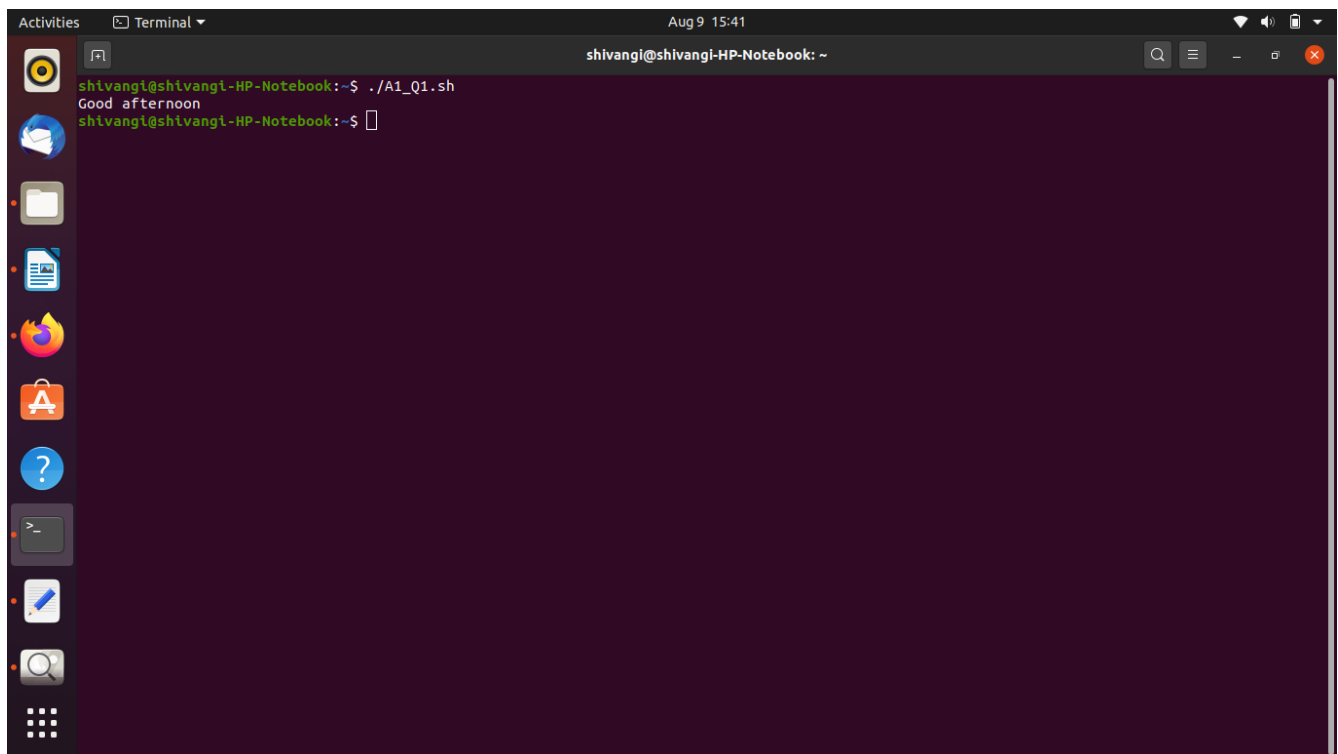
Assignment 1

Ques 1. [2 marks] Write a script which displays “Good morning”, “Good afternoon” or “Good evening”, on the monitor, depending on the system time of running the script.

Ans.

```
hour=`date +%H`{From the current date hour is extracted}
```

```
if [ $hour -lt 12 ]; then{If time is less than 12}
  echo Good morning
elif [ $hour -lt 18 ]; then
  echo Good afternoon
else
  echo Good evening
fi
```

A screenshot of a Linux terminal window. The window title is "Terminal" and it shows the date and time "Aug 9 15:41". The user is "shivangi" on a machine named "shivangi-HP-Notebook". The prompt is "shivangi@shivangi-HP-Notebook: ~". The user has entered the command "./A1_Q1.sh" and the terminal has outputted "Good afternoon". The terminal window is dark-themed with a purple background. On the left side, there is a vertical dock with various application icons including a file manager, a text editor, a web browser, and a terminal icon. The terminal window is open on top of the desktop environment.

```
shivangi@shivangi-HP-Notebook:~$ ./A1_Q1.sh
Good afternoon
shivangi@shivangi-HP-Notebook:~$
```

Ques 2. [5 marks] Write a bash script which reports the number of regular files in a given directory, with a possible filter on the file extension:

- Command Structure: `countfiles dirname [ext]`
- `dirname` must name a readable directory
- `ext`, if present, is used as a file extension in matching

Ans.

```
#!/bin/bash
for ext in jpg png gif tar pdf sh txt; do
  files=( *"$ext" )

  printf 'number of %s files: %d\n' "$ext" "${#files[@]}"

done

$(find -type f | sed -e 's/.*\.//'){files are listed with their extensions}
```

3. Given the following code:

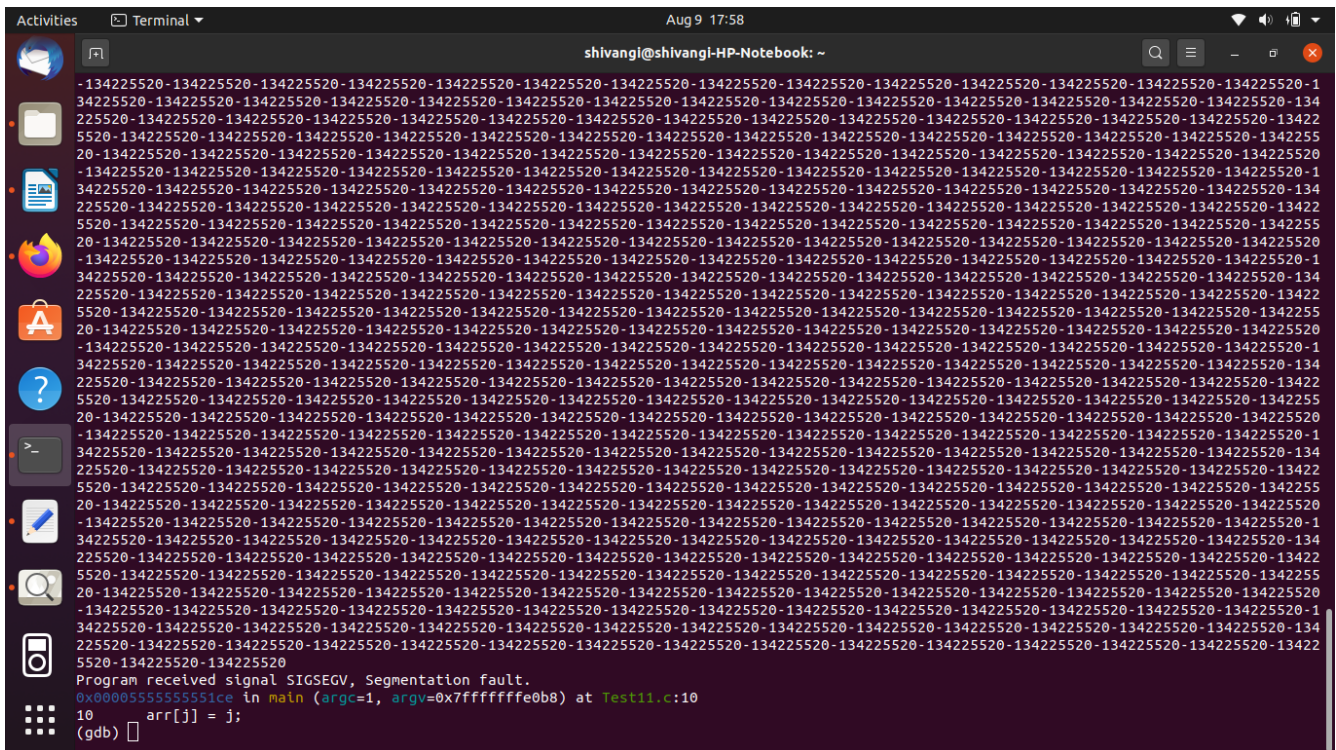
```
//main.c
#include <stdio.h>
#include "string.h"
int main(int argc, char* argv[]) {
    int i, j, k;
    int *p;
    int arr[500];
    char *str;
    for(j = 1; j <= 5000; ++j){
        arr[j] = j;
        printf("%d", *p);
    }
    str="System";
    modifyString(str);
    printf("Enter integer in 0..4999: ");
    scanf("%d", k);
    foo(arr, k);
}
int foo(int* c, int k) {
    printf("x[%d] = %d\n", k, c[k]);
}
void modifyString(char *s){
    *(s+3) = 'k';
}
```

i) [2 marks] Run this code using gdb debugging. Describe the steps you followed to find the error and to debug the code using gdb options

Following errors were resolved:

1.for(j = 1; j <= 499; ++j){//changed else would result in segmentation fault error

screenshot attached below



Segmentation fault

2.p=arr;//changed was first unitialized p

3.#include <string.h>//changed wrong use

4.void modifyString(char str[]);//Added function prototype

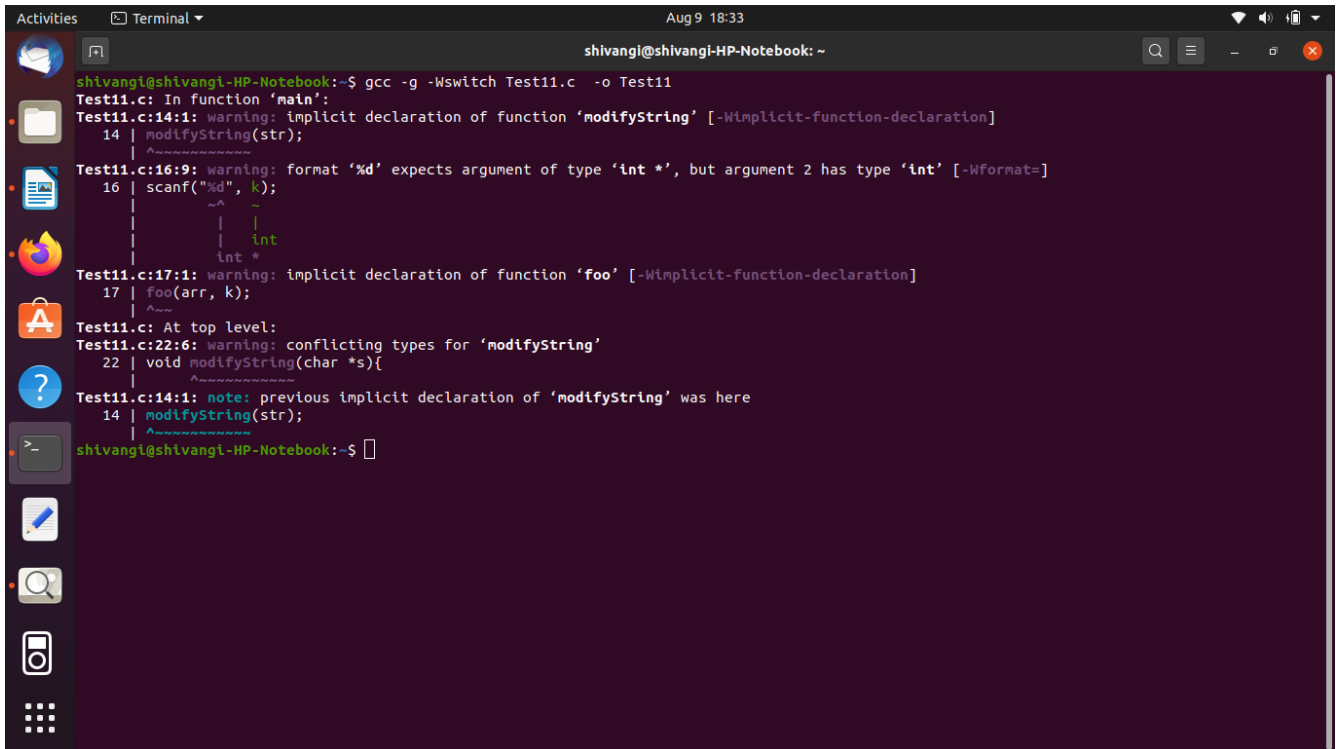
5.int foo(int* c, int k) ;//Added function prototype

6.char str[]="System";//First it was a string constant and we were changing the value hence resulted in error

7 scanf("%d", &k);//changed added &....

8.void modifyString(char str[]){//Changed to string

ii) [2 marks] Try compiling above code with the -Wall switch. What does it tell you? Is it useful?



```
shivangl@shivangl-HP-Notebook: ~  
shivangl@shivangl-HP-Notebook:~$ gcc -g -Wall Test11.c -o Test11  
Test11.c: In function 'main':  
Test11.c:14:1: warning: implicit declaration of function 'modifyString' [-Wimplicit-function-declaration]  
14 | modifyString(str);  
    |  
Test11.c:16:9: warning: format '%d' expects argument of type 'int *', but argument 2 has type 'int' [-Wformat=]  
16 | scanf("%d", k);  
    |         ^~  
    |         |  
    |         int *  
Test11.c:17:1: warning: implicit declaration of function 'foo' [-Wimplicit-function-declaration]  
17 | foo(arr, k);  
    |  
Test11.c: At top level:  
Test11.c:22:6: warning: conflicting types for 'modifyString'  
22 | void modifyString(char *s){  
    |  
Test11.c:14:1: note: previous implicit declaration of 'modifyString' was here  
14 | modifyString(str);  
    |  
shivangl@shivangl-HP-Notebook:~$
```

Explanation:

It displays warning related to possibility of incorrect code that's how it serves its utility.

Ex

Uninitialized use of variables, missing braces etc.

Ques 4. Enter the program listed below:

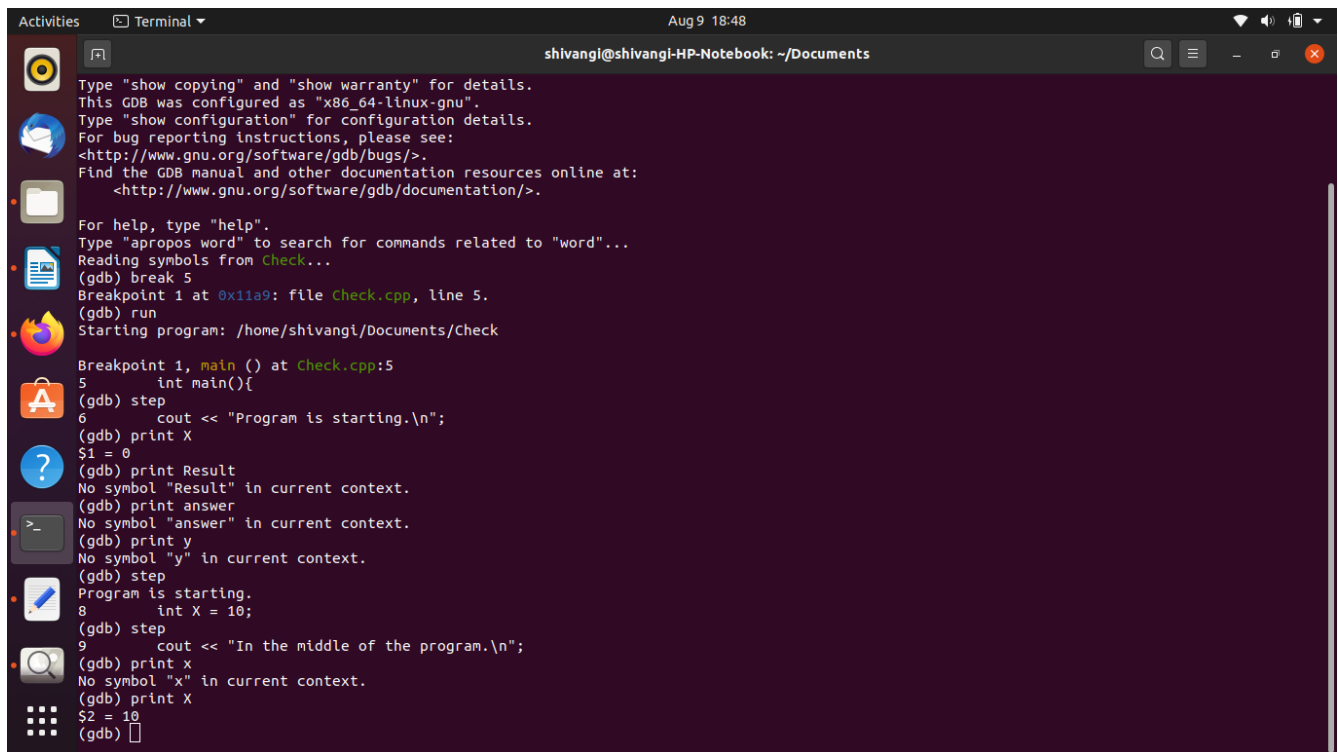
```
1] #include<iostream>
2] using namespace std;
3] int func(int);
4] const int X = 5;
5] int main(){
6] cout << "Program is starting.\n";
7] int result;
8] int X = 10;
9] cout << "In the middle of the program.\n";
10] result = func(X);
11] cout << "The result is " << result << ".\n";
12] }
13] int func(int y){
14] int answer;
15] cout << "In the function.\n";
16] answer = y;
17] answer += X;
18] int X = 20;
19] cout << "In the middle of the function.\n";
20] answer += X;
21] cout << "The function is exiting.\n";
22] return answer;
23] }
```

i) [3 marks] Hand simulate the program. Using the table given below, record the values of the variables X, result, answer, and y at the following lines in the program: 6,9,11,15,19,21. For some lines, some variables will not be defined, record that information instead of the variable's value.

Line Number	X	Result	Answer	Y
6	5	Variable s not defined	Variables not defined	Variables not defined
9	10	Garbage	Variables not defined	Variables not defined
11	5	35	Variables not defined	Variables not defined
15	5	Garbage	Garbage	10
19	20	Garbage	15	10
21	20	Garbage	35	10

ii) [3 marks] Using gdb and the break, step, and print variable commands record the values of the variables X, result, answer, and y at the following lines in the program: 6,9,11,15,19,21 in a different table. For some of these lines, you will get an error because the variable doesn't exist within the scope of that line. Do not go back and change the values in your first table.

Line Number	X	Result	Answer	Y
6	0	Not defined	Not defined	Not defined
9	10	0	Not defined	Not defined
11	32767	\000 <repeats 113 times>	-8320	32767
15	32767	000 <repeats 113 times>	-8320	10
19	20	000 <repeats 113 times>	15	10
21	20	000 <repeats 113 times>	35	10



```
shivangi@shivangi-HP-Notebook: ~/Documents
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from check...
(gdb) break 5
Breakpoint 1 at 0x11a9: file Check.cpp, line 5.
(gdb) run
Starting program: /home/shivangi/Documents/Check

Breakpoint 1, main () at Check.cpp:5
5      int main(){
(gdb) step
6      cout << "Program is starting.\n";
(gdb) print X
$1 = 0
(gdb) print Result
No symbol "Result" in current context.
(gdb) print answer
No symbol "answer" in current context.
(gdb) print y
No symbol "y" in current context.
(gdb) step
Program is starting.
8      int X = 10;
(gdb) step
9      cout << "In the middle of the program.\n";
(gdb) print x
No symbol "x" in current context.
(gdb) print X
$2 = 10
(gdb) 
```

ii) [3 marks] Answer the following questions:

1. On line 9, what is the value of the result and where did it come from? If you run the program several times, does the value of result at line 9 ever change?

Ans 1 value of the result is 10 and it takes the value from the local x.

NO value won't change on running the program several times.

2. What are the values printed for X on lines 9 and 15? Why are the printed values of X different?

Ans 2.X value on line 9:10

X value on line 15:32767(Garbage value)

3. Compare the values in Table your first and second tables? Were they different at any of the lines? If so explain the mistake you made during the hand simulation.

At line no 6:GDB takes default value as 0

At line 11 values are different since gdb doesnot uses scope resolution by default.