[Workshop Report-1]

Student Id : NP03A190051

Student Name : Rishab Sharma

Section : L5CG7

Lecturer : Jnaneshwor Bohara

Module Leader : Hiran Patel

Submitted on : 18th September, 2021

The following code prints out the value of an int variable and a string (char *) :

```
#include <stdio.h>
void main(int argc, char *argv[])
{
 int age = 10;
 char *name = "Hiran";
 printf("Hello %s, you are %d years old.", name, age);
}
```

1. Now modify the program so that it uses the command line arguments to supply name and age. i.e. it uses the argc and argv arguments/parameters.

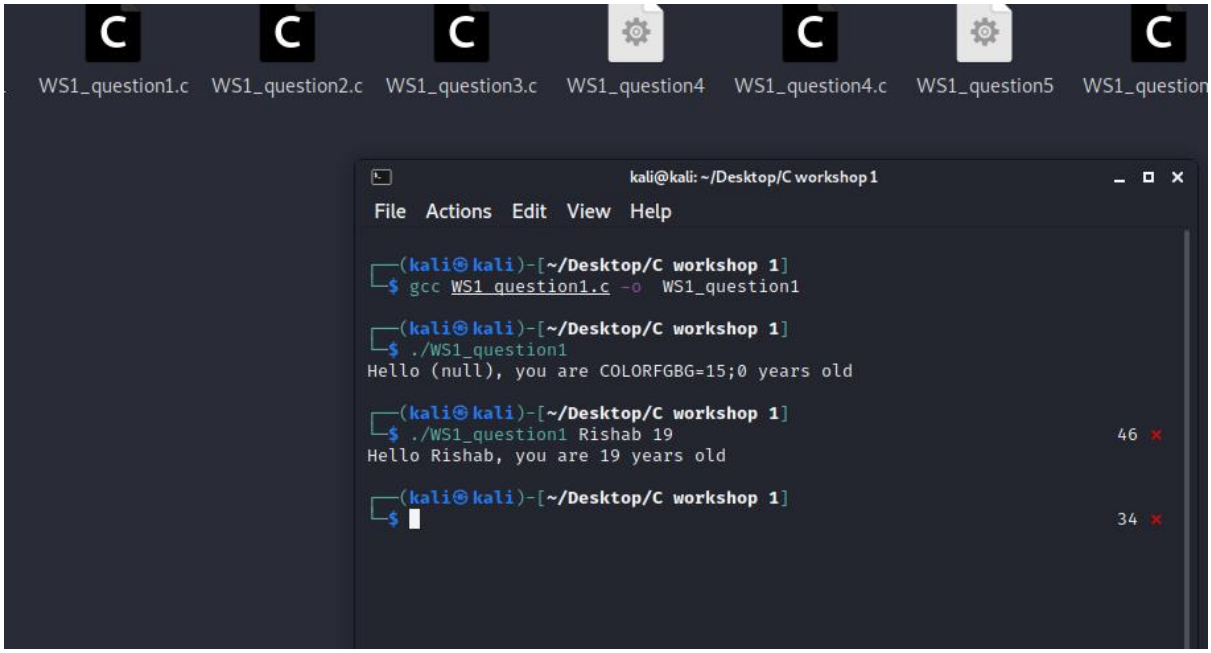
   When you run it, it should produce the following:

   ./myprog Jnaneshwar 100

   Hello Jnaneshwar, you are 100 years old.

= >

```
#include <stdio.h>
 #include <unistd.h>

void main (int argc, char *argv[])
 {
   printf("Hello %s, you are %s years old", argv[1], argv[2]);
}
```

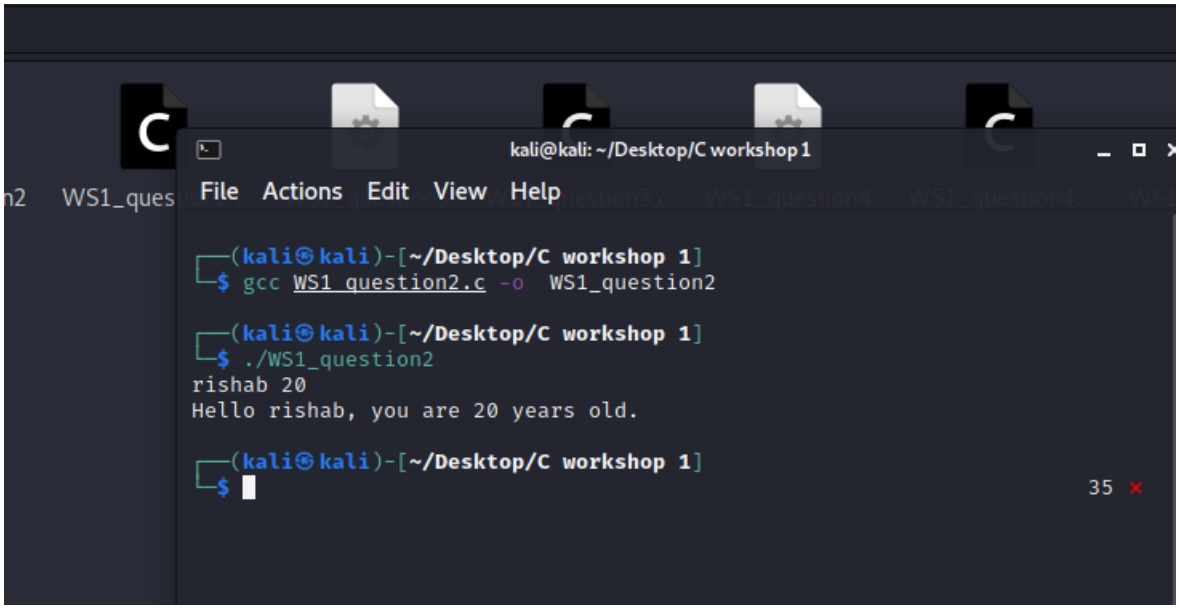


2. Now modify the program again so that it uses the scanf() function to get input from the user for the name and age.

   The following code count the integer variable n from 0 to 9 and prints out “Odd” if n is even and just the value of n if it is even:

   = > #include <stdio.h>

   ```
   void main()
   {
      char name [20];
      int age;
      scanf("%s", name);
      scanf("%d", &age);
      printf("Hello %s, you are %d years old.", name, age);
   }
   ```



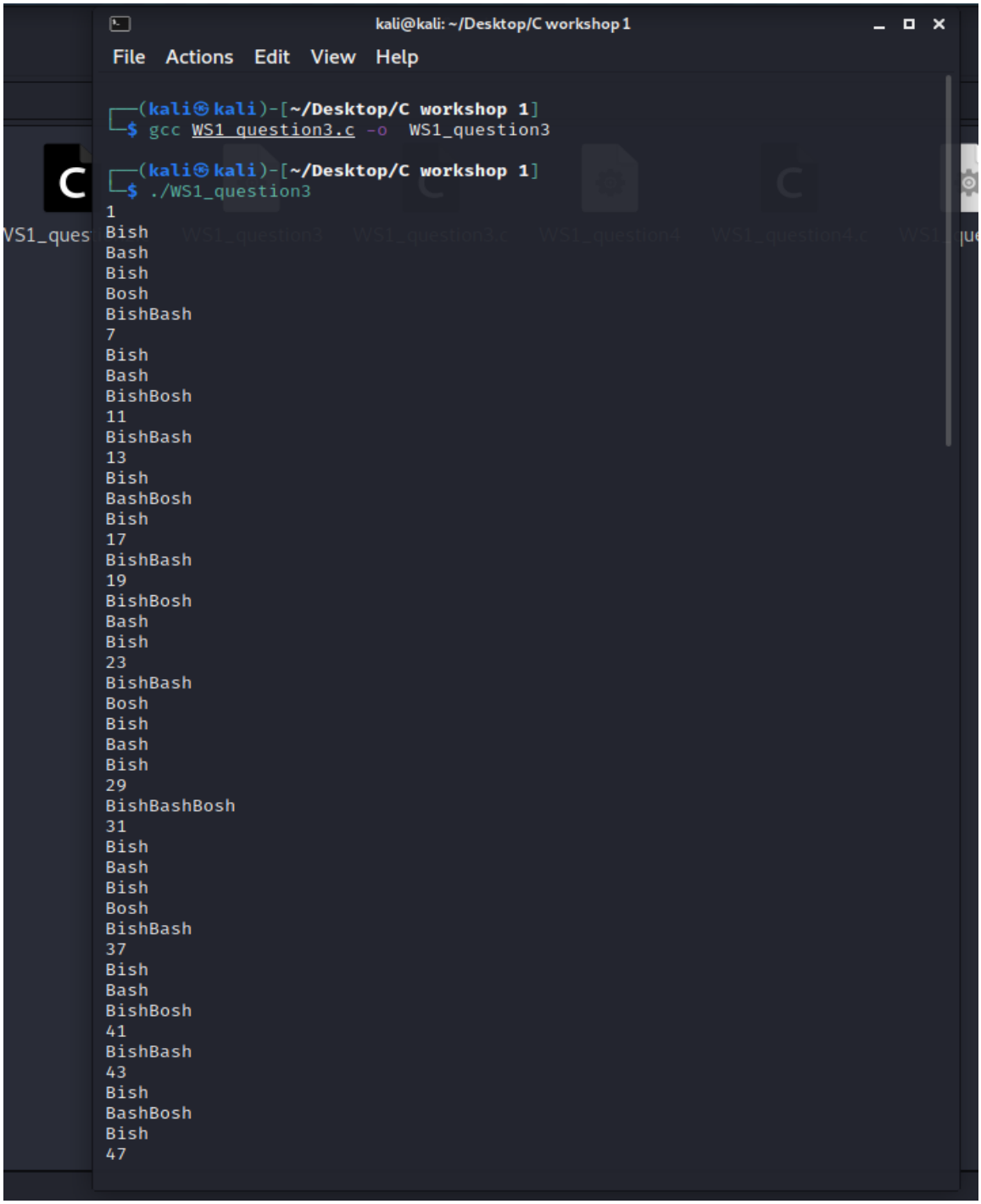
3. Now modify the program so that it counts the variable n from 1 to 100 and, if n is a multiple of 2 (eg. 2, 4, 6, etc), it would print out the word “Bish”, and if n is a multiple of 3 (eg. 3, 6, 9. 12 etc), it would print out the word “Bash”, and if n is a multiple of 5 (eg. 5, 10, 15 etc), it would print out the word “Bosh”.

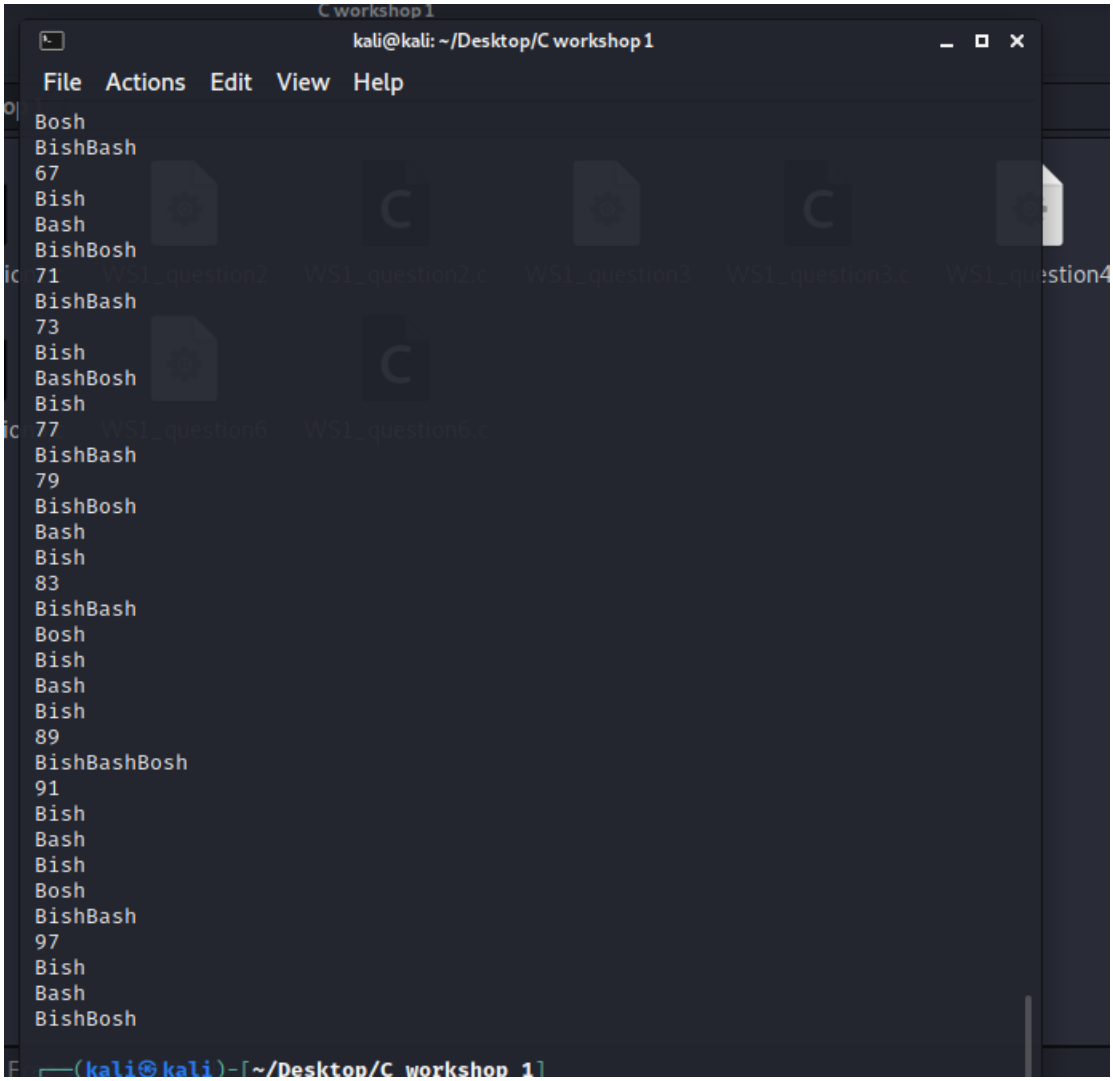
   However, if n is a multiple of 2 and 3 (eg. 6), it would print out the words “BishBash”, and if n is a multiple of 2 and 5 (eg. 10), it would print out the words “BishBosh”, and if n is a multiple of 3 and 5 (eg. 15), it would print out the words “BashBosh”. Finally, if n is a multiple of 2, 3 and 5 (eg. 30), it would print out the words “BishBashBosh”.

   = > #include <stdio.h>

```
void main (int argc, char *argv[]) {
    for (int n=1; n<=100; n++) {

        if(n % 2 == 0 && n % 3 == 0 && n % 5 == 0) {
            printf("BishBashBosh\n");
        }

        else if(n % 3 == 0 && n % 5 == 0) {
            printf("BashBosh\n");
        }

        else if(n % 2 == 0 && n % 5 == 0) {
            printf("BishBosh\n");
        }

        else if(n % 2 == 0 && n % 3 == 0) {
            printf("BishBash\n");
        }
```

```
        else if(n % 2 == 0) {
            printf("Bish\n");
        }

        else if(n % 3 == 0) {
            printf("Bash\n");
        }

        else if(n % 5 == 0) {
            printf("Bosh\n");
        }

        else if(n % 3 == 0) {
            printf("BishBash\n");
        }

        else {
            printf("%d\n", n);
        }
    }
}
```


kali@kali: ~/Desktop/C workshop 1
File Actions Edit View Help
(kali㉿kali)-[~/Desktop/C workshop 1]
$ gcc WS1_question3.c -o WS1_question3
(kali㉿kali)-[~/Desktop/C workshop 1]
$ ./WS1_question3
1
Bish
Bash
Bish
Bosh
BishBash
7
Bish
Bash
BishBosh
11
BishBash
13
Bish
BashBosh
Bish
17
BishBash
19
BishBosh
Bash
Bish
23
BishBash
Bosh
Bish
Bash
Bish
29
BishBashBosh
31
Bish
Bash
Bish
Bosh
BishBash
37
Bish
Bash
BishBosh
41
BishBash
43
Bish
BashBosh
Bish
47



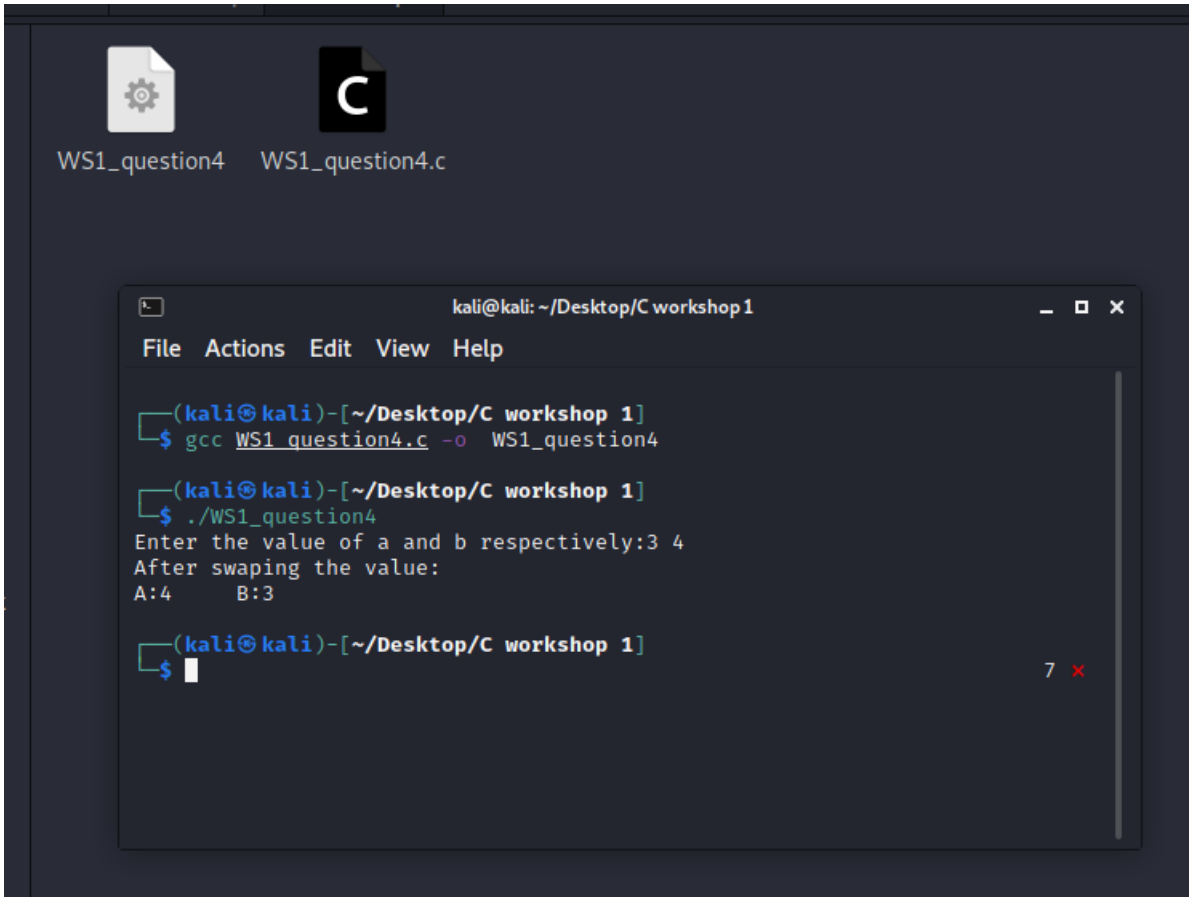
4. Now write a function called swap() that would swap the values of the variables a and b, when you call the swap() with the variables a and b as parameters. Please note, this exercise requires pointers.

   The following program fills an int array of size 10 and fills it with random numbers and prints them out:

   = > #include<stdio.h>

   ```
   void swap(int *a, int *b)
   {
       int temp;
       temp = *a;
       *a = *b;
       *b = temp;
   }
   ```

```
void main()
{
   int a,b;
   printf("Enter the value of a and b respectively:");
   scanf("%d%d",&a,&b);
   swap(&a,&b);
   printf("After swaping the value:\n");
   printf("A:%d\tB:%d",a,b);
}
```



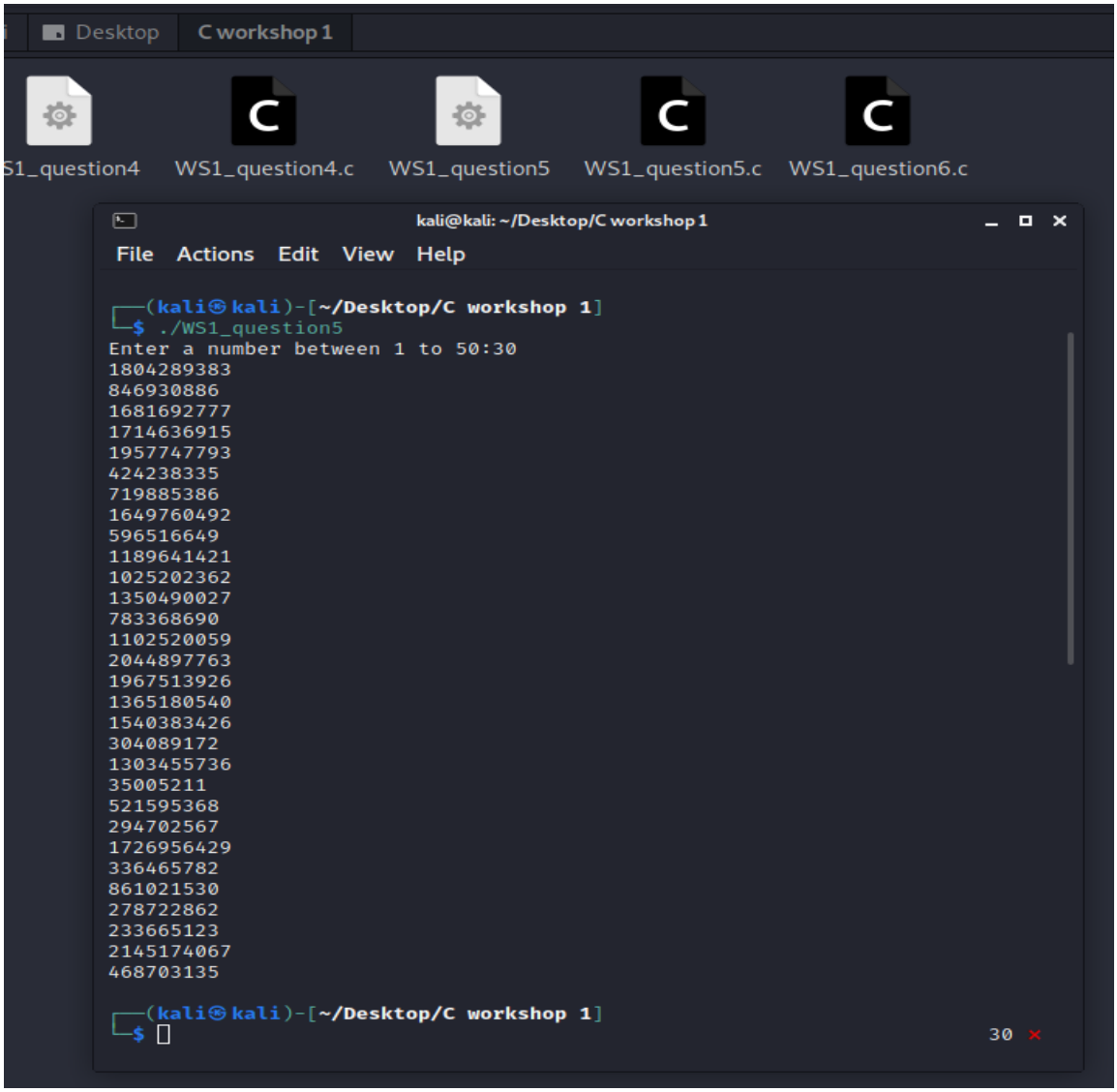
5. Now modify it to will ask the user for a number between 1 and 50, and then use the C function malloc() to allocate an int array of that size, fill it with random numbers and print out the value of each element of that array.

   The following code creates 2 threads in a program and counts to 10 in each thread :

   = > #include<stdio.h>
   #include<stdlib.h>

   ```
   void main(){
      int n,i;  // n is total number of items
      int *p;   //pointer variable
   ```

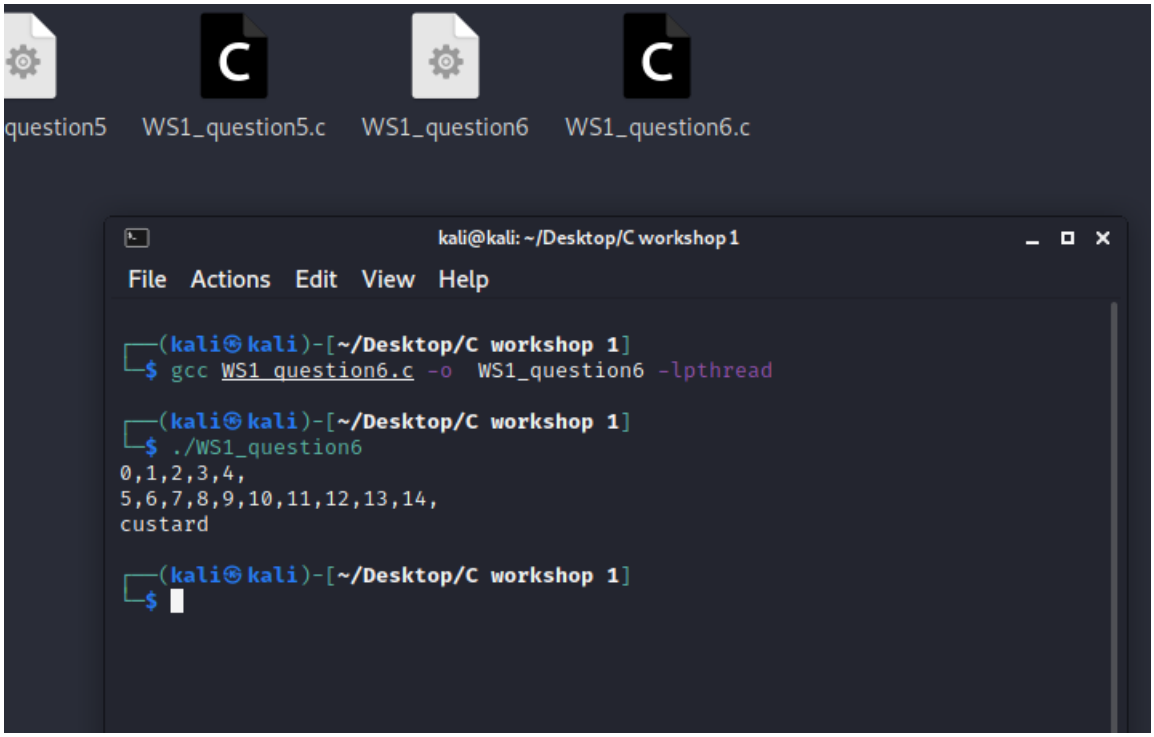
```
    printf("Enter a number between 1 to 50:");
    scanf("%d",&n);
    p = (int*)malloc(n * sizeof(int));
    for(i=0;i<n;i++){
        *(p+i) = rand();
    }
    for ( i = 0; i < n; i++)
    {

      printf("%d\n",*(p+i));

    }
}
```



6. Modify the program to accept a command line argument to specific the number of threads, and then create that many threads dynamically to run.

= > #include <stdlib.h>
#include <stdio.h>

```
int main() {
    int i;
    for(i=0;i<5;i++)
   {
        printf("%d,", i);
    }
    printf("\n");

    while(i<10)
   {
        printf("%d,", i);
        i++;
    }

    do {
        printf("%d,", i);
        i++;
      }
        while(i<15);
        printf("\n");


    if(i>13){
        printf("custard\n");
    } else
   {
        printf("gravy\n");
    }

    return EXIT_SUCCESS;
}
```


question5
WS1_question5.c
WS1_question6
WS1_question6.c
kali@kali: ~/Desktop/C workshop 1
File Actions Edit View Help
┌──(kali㉿kali)-[~/Desktop/C workshop 1]
└─$ gcc WS1_question6.c -o WS1_question6 -lpthread
┌──(kali㉿kali)-[~/Desktop/C workshop 1]
└─$ ./WS1_question6
0,1,2,3,4,
5,6,7,8,9,10,11,12,13,14,
custard
┌──(kali㉿kali)-[~/Desktop/C workshop 1]
└─$