University of Wolverhampton
School of Mathematics and Computer Science

Student Number:
Name:

6CS005 High Performance Computing Week 1 Workshop

Revision on C and Multithreading

Tasks – Basic C Syntax

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.

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(int argc, char *argv[])
{
  for(int n =0; n <10; n++){
    if(n % 2 == 1){
      printf("%d is Odd\n", n);
    }
    else{
      printf("%d\n", n);
    }
  }
}
```

When you run the program, it should output the following:

```
0
1 is Odd
2
3 is Odd
4
5 is Odd
6
7 is Odd
8
9 is Odd
```

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”.

When you run the program, it will produce something like this:

```
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
```

The following code swaps the values of the two variables a and b ::

```
#include <stdio.h>

void main(int argc, char *argv[])
{
  int a = 3;
  int b = 4;
  int temp = 0;

  printf("a is %d and b is %d\n", a, b);

  temp = a;
  a = b;
  b = temp;

  printf("a is now %d and b is now %d", a, b);
}
```

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>
#include <stdlib.h>

void main(int argc, char *argv[])
{
  int numbers[10];

  for (int i=0; i < 10; i++){
    numbers[i] = rand();
    printf("%d is %d\n", i, numbers[i]);
  }
}
```

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

void *threadA(void *p){
  for(int i=0; i<10; i++){
    printf("Thread ID %ld: i=%d\n", pthread_self(), i);
    usleep(1000);
```

```
  }
}
void *threadB(void *p){
  for(int i=0; i<10; i++){
    printf("Thread ID %ld: i=%d\n", pthread_self(), i);
    usleep(1000);
  }
}
void main(){
  pthread_t thrID1, thrID2;
  pthread_create(&thrID1, NULL, threadA, NULL);
  pthread_create(&thrID2, NULL, threadB, NULL);
  pthread_join(thrID1, NULL);
  pthread_join(thrID2, NULL);
}
```

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.