CO222: Programming Methodology

Problem Solving and Coding Workshop

Part A: Reading and Understanding Correct Code

Assuming that the following C programs are valid and they execute without errors, indicate clearly and exactly what output will be printed.

Program-A01

```
#include <stdio.h>

int main( void )
{
	int x=1;
	int y=0;

	while( x < 100 ) {
		x = x + 8 * y++;
		printf("%d\n", x);
	}
	return 0;
}
```

Program-A02

```
#include <stdio.h>
int main( void )
{
	int *p, *q;
	int x, y;

	x = 7;
	y = 8;

	q = &x;
	*q = 10;
	p = q;

	y = *p + *q;
	*p = x + y;

	printf("x=%d, y=%d\n",x,y);

	return 0;
}
```

Program-A03

```
#include <stdio.h>
int df( int n ){
	if( n < 2 ){
		return(1);
	}else{
		return(n * df(n-2));
	}
}

int main( void ){
	printf("5!! = %d\n", df(5));
	printf("6!! = %d\n", df(6));
	return 0;
}
```

Program-A04

```
#include <stdio.h>
void f( char *s )
{
	if( !*s ) {
		return;
	}
	f( s+1 );
	putchar( *s );
}

int main(void)
{
	f("kernighan");
	putchar('\n');

	return 0;
}
```

Program-A05

```
#include<stdio.h>
void print_edge( int k ){
      int i;
      for( i=0; i < k; i++ ) {
            printf( "+-" );
      }
      printf( "+\n" );
}

void print_legs( int k ){
      int i;
      for( i=0; i < k; i++ ) {
            printf( "| " );
      }
      printf( "|\n" );
}

int main(void ){
int k;
      for( k=0; k < 4; k++ ) {
            print_edge( k );
            print_legs( k );
      }
      print_edge( k );
      return 0;
}
```

Program-A06

```
#include<stdio.h>
enum {false,true};

int main()
{
        int i=1;
        do{
                printf("%d\n",i);
                i++;
                if(i < 15)
                        continue;
        }while(false);
        return 0;
}
```

Program-A07

```
#include <stdio.h>
int a = 1;
static int b = 1;

int f( int c )
{
      static int d = 1;
      int e = 0;

      a++;
      b += d;
      c = c + 2;
      d = d + a - b + c;
      e = e + 2*d + 1;
      return(e+2);
}

int main( void )
{
      int a, d;
      a = 3;

      for( d=0; d < 3; d++ ) {
            printf("%d\n", f(a));
      }
      printf("%d\n", a );
      printf("%d\n", b );
      printf("%d\n", d );

      return 0;
}
```

Program-A08

```
#include <stdio.h>

int main()
{
      int i = 43;
      printf("%d\n", printf("%d", printf("%d", i)));
      return 0;
}
```

Program-A09

```
#include<stdio.h>

int myFunc1(unsigned int x)
{
    int c = 0;
    while(x)
    {
        c++;
        x = x&(x-1);
    }
    return c;
}

int myFunc2(unsigned int x)
{
      static unsigned int mask[] = {0x55555555, 0x33333333, 0x0F0F0F0F,
                                    0x00FF00FF, 0x0000FFFF};
      int i ;
      int shift ;
      for ( i =0, shift =1; i < 5; i ++, shift *= 2)
            x = (x & mask[i])+ ( ( x >> shift) & mask[i]);
      return x;
}

int main(){
      int i;
      for(i = 0; i < 16; i++)
            printf("%d -> %d -> %d\n", i, myFunc1(i), myFunc2(i));

      return 0;
}
```

Program-A10

```
#include <stdio.h>

void myFun(int *x , int *y)
{
      int tmp = *x ;
      *x = *y ;
      *y = tmp;
}

int main()
{
      int a = 1, b = 2;
      myFun(&a, &b);
      printf("%d %d\n", a, b);
      return 0;
}
```

Program-A11

```
#include <stdio.h>

void f(int n)
{
      printf("F");
      if (n != 0)
            f(n-1);
}

int main()
{
      int x;
      x = 3;
      f(x);
      printf("MAIN");
      return 0;
}
```

Program-A12

```
#include <stdio.h>

int main()
{
      int i;
      i = 10;
      printf("i : %d\n",i);
      printf("5 OR i is: %d\n", 5 || i++);
      printf("i : %d\n",i);
      return 0;
}
```

Program-A13

```
#include <stdio.h>

int main()
{
      int i;
      i = 0;
      printf("i : %d\n",i);
      printf("5 OR i is: %d\n", 5 && i++);
      printf("i : %d\n",i);
      return 0;
}
```

Part B: Reading and Identifying Errors in Code

Read and identify the errors/mistakes in the following C code (the errors can be compile time or runtime). At first you should try to identify the errors without using a compiler.

Program-B01

```
#include <stdio.h>

int main()
{
  int n;

  printf("Enter a number:\n");
  scanf("%d\n",n);

  printf("You entered %d \n",n);

  return 0;
}
```

Program-B02

```
#include <stdio.h>

int main()
{
      int cnt = 5, a;

      do{
            a  /= cnt;
      } while (cnt--);

      printf ("%d\n", a);

      return 0;
}
```

Program-B03

```
#include <stdio.h>

int fiveTimes(int a)
{
  int t;
  t = a<<2 + a;
  return t;
}

int main()
{
  int x = 2;
  printf("Five times of %d is %d\n", x, fiveTimes(x));

  return 0;
}
```

Program-B04

```
#include<stdio.h>

void concat(char* f, char* s)
{
      while(*f++);

      while((*f++ = *s++));
}

int main(){
      char s1[] = "Hello ";
      char s2[] = "World";

      concat(s1,s2);

      printf("%s",s1);

      return 0;
}
```

Program-B05

```
#include <stdio.h>

int max(int x, int y)
{
      (x > y) ? return x : return y;
}

int main()
{
      int a = 10, b = 20;
      printf("max(%d,%d) = %d", a, b, max(a,b));
      return 0;
}
```

Part C: Coding exercises

C01. Sorting an array (simple sort)

Consider an array of integers of size n, whose values are set randomly using the rand() function. Write a C code that sorts the elements from the smallest to the largest value. One will proceed as follow:

1) search for the minimum of the elements 0 to n-1. Swap the minimum and first element.
2) Search for the minimum of the elements 1 to n-1. Swap the minimum and second element.
3) ...
4) repeat until the array is sorted.

C02. Concatenation of two sorted arrays

Consider X and Y, two sorted arrays of integers. Write a C code that concatenates the two arrays into one array Z (sorted) which contains the elements of X and Y.

C03. Largest Palindrome number

A palindromic number reads the same both ways. The largest palindrome made from the product of two 2-digit numbers is 9009 = 91 × 99.
Find the largest palindrome made from the product of two 3-digit numbers.

C04. Rotate an array

Write a function that rotates a list by k elements. For example [1,2,3,4,5,6] rotated by two becomes [3,4,5,6,1,2]. Try solving this without creating a copy of the list.

C05. Print string in a frame

Write a function that takes a list of strings and prints them, one per line, in a rectangular frame. For example, the list ["Hello", "World", "in", "a", "frame"] gets printed as:

```
*********
* Hello *
* World *
* in    *
* a     *
* frame *
*********
```

C06. Cartesian to Polar coordinates

Write a program which accepts Cartesian coordinates x and y, and prints its polar coordinates form i.e. r and theta (degrees only).

For example:
if user input x=3 and y=5 then it should print r and theta as
5.8309
59.0362
if user input x = 20 and y = 34 then it should print r and theta as
39.4462
59.5345

C07. Consecutive digits

Write a code for checking whether the given number has all of its digits consecutive. To understand better two examples are given below.

Input: 123
Output: the given number has all of its digits consecutive

Input: 9432
Output: the given number do not have all of its digits consecutive

C08. Printing a Number Triangle

Write a C program that takes a number from the user and prints a triangle based on the number as shown in the following examples (inputs and outputs).

```
$./c08

Enter the size:

3
        1
     1  2
  1  2  3
```