

Fundamentals of Computing and Programming

Class Assignment 1. 20 marks, Time Limit 30 minutes

September 12, 2022

1. What does the following function print: [2]

```
void f(){
    int n=100, m=200;
    n = (m=50+2)+2;
    printf("%d %d",n,m);
}
```

2. Carefully read the function below. **Mention two mistakes** in the function: [2]

```
void g(){
    int n;
    scanf("%d", n );
    m = n+1;
    printf("%d", m) ;
}
```

3. Complete the following *main* function skeleton that **reads three integers** x , y and z and **prints the sum of the squares** of these numbers. [2]

```
void main(){
    int ... ;    // Declares the required variables
    scanf(...); // Reads into the variables
    printf(...); // Computes and prints the desired value
}
```

4. (a) Write out a function named *mydiv* to take **two integer parameter values** *x* and *y* and **print the value of the floating point division** of *x* by *y*. [2]
- (b) Illustrate how this function is used by writing a *main()* function that calls *mydiv* with some actual parameter values. [2]
5. Read the code below:

```
void test_1(int x, int y){
    x=20;
    y=10;
    printf("\n IN : %d %d", x, y)
}
void main(){
    int a = 400;
    int b = 800;
    int x = 10;
    int y = 20;
    test_1(a, b);
    printf("\n OUT: %d %d %d %d", a, b, x, y);
}
```

Which of these is the correct output: [2]

- (a) IN : 800 400
OUT: 400 800 20 10
- (b) IN : 10 20
OUT: 400 800 20 10
- (c) IN : 20 10
OUT: 400 800 10 20
- (d) IN : 20 10
OUT: 400 800 20 10

6. What is printed by the given program: [2]

```

void main(){
    int n=100,m=200;
    int *p1; int *p2;
    p1=&n; p2=&m;
    p1=p2;
    *p1=0;
    printf("%d %d",n,m);
}

```

7. Write a function named *myadd* that takes two parameters *x* and *py*. *x* is an integer and *py* is a pointer to another integer location. The function adds the value of *x* to the contents of the integer pointed to by *py*. [2]
8. There are four integer variables *a, b, c, d* a function. The programmer requires that the four integers *a, b, c, d* **should not be** in strict increasing or strict decreasing order. Write a single **if** statement that checks that the four variables are neither in the order $a < b < c < d$ nor in the order $a > b > c > d$. If the condition is violated print "Bad", else print "Good". [2]
9. Write a function named *read3* which has no parameters. It repeatedly reads an integer input from the user and simply checks if that given input is divisible by 3. The loop keeps repeating until the input read is divisible by 3. [2]