Class Test – 1 – Section D9

1. Write a C program that asks the user to enter a 2-digit number, then prints the number with its digits reversed. [Here consider ONLY a two-digit number and nothing else] **4**

2. As a programmer, if you want a user to give you any character value as input, which function will you need?
   Write only the function syntax where that character input gets stored in a character variable, where the variable must be your name. **1+2**

3. Are there any mistakes in the following code? Write Yes or No. If yes, write down the correct C code. **3**

```
define stdio>
float Main(1)
[ int variable = 5:
  print("%d\n", &variable);
]
```

4. Supply the brackets in such a way that a compiler would compile these following expressions: **2**
   i) m * -n / p – q
   ii) a / b % c / d

5. Write only the outputs of the following code snippets according to the statements given with each. **3**

i)
```
i = 4; j = 5; k = 6;
printf("%d ",i++ - j++ + --k);
printf("%d %d %d", i, j, k);
```

ii)
```
i = 6; j = 3 + --(--i) * 2;
printf("%d %d", i, j);
```

Class Test – 1 – Section D10

1. Write a C program that asks the user to enter a 2-digit number, then prints the number with its digits reversed. [Here consider ONLY a two-digit number and nothing else] **4**

2. As a programmer, if you want a user to give you any character value as input, which function will you need?
   Write the function syntax where that character input gets stored in a character variable, which is your name. **1+2**

3. Are there any mistakes in the following code? Write Yes or No. If yes, write down the correct C code. **3**

```
define stdio>
float Main(1)
[ int variable = 5:
  print("%d\n", &variable);
]
```

4. Supply the brackets in such a way that a compiler would compile these following expressions: **2**
   i) m * -n / p – q
   ii) a / b % c / d

5. Write only the outputs of the following code snippets according to the statements given with each. **3**

i)
```
i=3; j=4; k=5;
printf("%d ",i++ - j++ + --k);
printf("%d %d %d", i, j, k);
```

ii)
```
i = 7; j=3 + --i * 2;
printf("%d %d", i, j);
```