1. **Structures, Unions and Enumerations all of them let you_____**

- ○ Define new structures
- ○ Define new data values
- ○ Define new pointers
- ◉ Define new data types

2. What is the output of this program?

```
#include <stdio.h>
struct student
{
  int no = 5;
  char name[20];
};
void main()
{
  struct student s;
  s.no = 8;
  printf("hello");
}
```

- ◉ Compile Time Error
- ○ Nothing
- ○ Varies
- ○ Hello

3. What is the output of this program?

```
#include <stdio.h>
struct student
{
  char *c;
};
void main()
{
  struct student s[2];
  printf("%d", sizeof(s));
}
```

- ○ 2
- ◉ 16
- ○ 4
- ○ 8

4. **Size of a union is determined by size of the.**

- ○ First member in the union
- ○ Sum of the sizes of all members

- ○ Last member in the union C. Biggest member in the union
- ◉ Biggest member in the union

5. What will be the output of the following pseudocode?

```
input m=9,n=6
m=m+1
n=n-1
m=m+n
if(m>n)
   print m
else
   print n
```

- ○ 10
- ○ 6
- ○ 5
- ◉ 15

6. What is the output for the following code, if the value of the num is 25?

```
x = 0
Input n
Repeat till(n != 1) {
   if(n % 5 == 0) {
      n /= 5;
   }
   else if(n % 3 == 0) {
      n /= 3;
   }
   else if(n % 2 == 0) {
      n /= 2;
   }
   else {
      Print("Not an ugly number");
      x = 1;
      break;
   }
}
if(x == 0) {
   Print("Ugly number");
}
```

- ○ Not an ugly number Ugly number
- ○ Ugly number Not an ugly number
- ○ Not an ugly number
- ◉ Ugly number

7.

What will be the output of the following pseudocode?

```
Integer a, b
Set a = 3, b = 5
if (a & (b + 1))
a = a + a
End if
Print a + b
```

[Note: If(x) gets executed if the value inside if(), i.e., x is not zero
&: bitwise AND - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1 corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0]

- (●) **11**
- ( ) **30**
- ( ) **8**
- ( ) **7**

8. **What will be the output of the following pseudocode?**

```
Integer a, b
Set a = 10
Set b = a + a
if (b > a && 0)
   b = b - a
   b = b mod a
End if
if (b > a || 0)
   b = b + a
   b = b + a
End if
Print b
```

[Note- mod finds the remainder after the division of one number by another. For example, the "5 mod 2" would evaluate to 1 because 5 divided by 2 leaves a quotient of 2 and a remainder of 1]
[Note-&&: Logical AND - The logical AND operator (&&) returns the Boolean value true(or 1) if both operands are true and return false (or 0) otherwise
&: bitwise AND - The bitwise AND operator (&) compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0]

- ( ) **50**
- ( ) **10**
- (●) **40**
- ( ) **20**

9. **What will be the output of the following pseudocode for a = 4, b = 3?**

```
1. Integer funn(Integer a, Integer b)
2. a=b
3. b=a
```

```
4. if(a>b || b>a)
5.          return a + b
6. End if
7.          return a-b+10
8. End function funn()
```

- ○ 22
- ○ 12
- ○ 8
- ◉ 10

10. What will be the output of the following pseudocode?

```
1. Integer a, b, c
2. Set a = 1, b = 12, c = 2
3. if(1 < 2)
4.          if(1 || 1)
5.                    a = a ^ c
6.          End if
7.          a = a ^ b
8. End if
9. Print a - b + c
```

[Note- ||: Logical OR - The logical OR operator (||) returns the Boolean value TRUE (or 1) if either or both operands are true and return FALSE (or 0) otherwise.
^ is the bitwise exclusive OR operator that compares each bit of its first operand to the corresponding bits of its second operand. If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.]

- ◉ 5
- ○ 10
- ○ 19
- ○ 2