**Python Worksheet 1**

**Q1 – C)%**

**Q2-B)0**

**Q3-C)24**

**Q4-A)2**

**Q5-D)6**

**Q6-C)** the finally block will be executed no matter if the try block raises an error or not.

**Q7-A)** It is used to raise an exception.

Q8-C) in defining a generator

Q9-A)_abc

C)abc2

Q10-A)yield

B)raise

Q11-n = 23

fact = 1

```
for i in range(1, n+1):
   fact = fact * i

print("The factorial of 23 is : ", end="")
print(fact)
```

```
The factorial of 23 is : 25852016738884976640000
```

Q12-num = 11

```
# If given number is greater than 1
if num > 1:
   # Iterate from 2 to n / 2
   for i in range(2, int(num/2)+1):
      # If num is divisible by any number between
      # 2 and n / 2, it is not prime
      if (num % i) == 0:
```

```
        print(num, "is not a prime number")
        break
    else:
        print(num, "is a prime number")
else:
    print(num, "is a composite number")
```

```
11 is a prime number
```

Q13-# function which return reverse of a string

```
def isPalindrome(s):
    return s == s[::-1]


# Driver code
s = "malayalam"
ans = isPalindrome(s)

if ans:
    print("Yes")
else:
    print("No")
```

```
Yes
```

Q14-

```
double findHypotenuse(double side1, double side2)
{
    double h = sqrt((side1 * side1) + (side2 * side2));
    return h;
}

// Driver code
int main()
{
    int side1 = 3, side2 = 4;
    cout << fixed << showpoint;
    cout << setprecision(2);
```

```
    cout << findHypotenuse(side1, side2);
}
```

Q15- def factorial(n):

```
    if n==1:
        return n
    else:
        return n* factorial(n-1)
print(factorial(4))
```

24