**ID prueba: 1**

**Descripción de la prueba: Prueba de modulo “factorial ”, la función dentro del paquete de programas. Realiza una operación para sacar el factorial de un número.**

**Supuestos y condiciones previas: Las funciones están separas por modulo y están validadas.**

**Datos de prueba: un numero positivo**

**7**

**Pasos a ejecutar:**

1. **Importar la función a probar**
2. **Crear variable con el valor n= 7**
3. **Ejecutar la función con el valor anterior**
4. **Comparar el resultado con el resultado esperado resultado = 5,040**

**Resultado real y condiciones posteriores:**

**El resultado de esta prueba fue obtener como resultado el factorial de un numero positivo, retornando el resultado que es: 5,040**

**Estado de la prueba:**

**Aprobado**

**ID prueba: 2**

**Descripción de la prueba: Prueba de modulo “factorial”, la función dentro del paquete de programas. Realiza una operación para sacar el factorial de un número.**

**Supuestos y condiciones previas: Las funciones están separas por modulo y están validadas.**

**Datos de prueba: un numero negativo debe ser rechazado poniendo a prueba que solo se admiten números positivos**

**-8**

**Pasos a ejecutar:**

1. **Importar la función a probar**
2. **Crear variable con el valor n= -8**
3. **Ejecutar la función con los valores anteriores por separado**
4. **Comparar el resultado con el resultado esperado resultado = -40,320**

**Resultado real y condiciones posteriores:**

**El resultado de esta prueba fue obtener como resultado el factorial de un numero negativo, retornando el resultado que es: “deben ser números positivos”**

**Estado de la prueba:**

**Aprobado**

**ID prueba: 3**

**Descripción de la prueba: Prueba de modulo “combinación”, la función dentro del paquete de programas. Realiza una operación para obtener el resultado de una combinación.**

**Supuestos y condiciones previas: Las funciones están separas por modulo y están validadas.**

**Datos de prueba: un par de números positivos**

**8 y 3**

**Pasos a ejecutar:**

1. **Importar la función a probar**
2. **Crear variables con el valor n= 8 y r= 3**
3. **Ejecutar la función con los valores anteriores**
4. **Comparar el resultado con el resultado esperado resultado = 56**

**Resultado real y condiciones posteriores:**

**El resultado de esta prueba fue obtener como resultado la combinación de los números, retornando el resultado 56**

**Estado de la prueba:**

**Aprobado**

**ID prueba: 4**

**Descripción de la prueba: Prueba de modulo “combinación”, la función dentro del paquete de programas. Realiza una operación para obtener el resultado de una combinación.**

**Supuestos y condiciones previas: Las funciones están separas por modulo y están validadas.**

**Datos de prueba: un par de números negativos**

**-8 y -3**

**Pasos a ejecutar:**

1. **Importar la función a probar**
2. **Crear variables con el valor n= 8 y r= 3**
3. **Ejecutar la función con los valores anteriores**
4. **Comparar el resultado con el resultado esperado resultado = “n y r deben ser números no negativos”**

**Resultado real y condiciones posteriores:**

**El resultado de esta prueba fue obtener como resultado el mensaje “r y r deben ser números no negativos”**

**Estado de la prueba:**

**Aprobado**

**ID prueba: 5**

**Descripción de la prueba: Prueba de modulo “opera”, la función dentro del paquete de programas. Realiza una comprobación para no ingresar valores tipo string**

**Supuestos y condiciones previas: Las funciones están separas por modulo y están validadas.**

**Datos de prueba: un numero y una palabra**

**9 y hola**

**Pasos a ejecutar:**

1. **Importar la función a probar**
2. **Crear variables con el valor n= 9 y p= hola**
3. **Ejecutar la función con los valores anteriores**
4. **Comparar el resultado con el resultado esperado resultado = “n y p deben ser de tipo string”**

**Resultado real y condiciones posteriores:**

**El resultado de esta prueba fue obtener como resultado el mensaje “n y p deben ser de tipo string”**

**Estado de la prueba:**

**Aprobado**

**ID prueba: 6**

**Descripción de la prueba: Prueba de modulo “opera”, la función dentro del paquete de programas. Realiza una comprobación para no ingresar valores tipo float**

**Supuestos y condiciones previas: Las funciones están separas por modulo y están validadas.**

**Datos de prueba: un numero entero y uno con punto decimal**

**5 y 5.5**

**Pasos a ejecutar:**

1. **Importar la función a probar**
2. **Crear variables con el valor n= 5 y r=5.5**
3. **Ejecutar la función con los valores anteriores**
4. **Comparar el resultado con el resultado esperado resultado = “n y r deben ser números enteros no de punto flotante”**

**Resultado real y condiciones posteriores:**

**El resultado de esta prueba fue obtener como resultado el mensaje “n y r deben ser números enteros no de punto flotante”**

**Estado de la prueba:**

**Aprobado**

**ID prueba: 7**

**Descripción de la prueba: Prueba de modulo “opera”, la función dentro del paquete de programas. Realiza una comprobación para un numero no debe ser mayo que el segundo**

**Supuestos y condiciones previas: Las funciones están separas por modulo y están validadas.**

**Datos de prueba: un numero mayor y uno menor**

**5 y 9**

**Pasos a ejecutar:**

1. **Importar la función a probar**
2. **Crear variables con el valor n= 5 y r= 9**
3. **Ejecutar la función con los valores anteriores**
4. **Comparar el resultado con el resultado esperado resultado = “n debe ser mayor que r”**

**Resultado real y condiciones posteriores:**

**El resultado de esta prueba fue obtener como resultado el mensaje “n debe ser mayor que r”**

**Estado de la prueba:**

**Aprobado**

**ID prueba: 8**

**Descripción de la prueba: Prueba de modulo “opera”, la función dentro del paquete de programas. Realiza una operación de combinación**

**Supuestos y condiciones previas: Las funciones están separas por modulo y están validadas.**

**Datos de prueba: dos números que cumplen con las condiciones**

**8 y 3**

**Pasos a ejecutar:**

1. **Importar la función a probar**
2. **Crear variables con el valor n= 8 y r= 3**
3. **Ejecutar la función con los valores anteriores**
4. **Comparar el resultado con el resultado esperado, resultado = 56**

**Resultado real y condiciones posteriores:**

**El resultado de esta prueba fue obtener como resultado= 56**

**Estado de la prueba:**

**Reprobado**