Test Assignment

C++

Name Surname (12345678)

February 25, 2026

Question 1

Returning multiple values is possible using tuples

program.cpp

```cpp
#include <tuple>
#include <iostream>

std::tuple<int,int,int> somefunc()
{
        return {1, 2, 3};
}

int main()
{
        auto& [a, b, c] = somefunc();
        std::cout << a << b << c << std::endl;
        return 0;
}
```

Output

```
123
```

Question 2

Question 2.1

Parsing simple binary operations with a case statement

program.cpp

```cpp
#include <iostream>

int main()
{
    while(true)
    {
        double a, b, result;
        char op;
        std::cin >> a >> op >> b;
        switch (op)
        {
        case '+':
            result = a + b;
            break;
        case '-':
            result = a - b;
            break;
        case '*':
            result = a * b;
            break;
        case '/':
            result = a / b;
            break;
        }
        std::cout << a << ' ' << op << ' ' << b << " = " << result << std::endl;
    }

}
```

Output

```
1 + 1
1 + 1  = 2
5 - 6
5 - 6 = -1
2 * 3
2 * 3 = 6
5 / 2
5 / 2 = 2.5
C^
```

Question 2.2

program.cpp

```cpp
#include <iostream>
double power(double base, int exponent)
{
    if(exponent)
    {
        int exp = (exponent > 0)? exponent : -exponent;
        double res = 1;
        for(int i = 0; i < exp;i++)
        {
            res *= base;
        }
        return (exponent > 0)? res : 1/res;
    }
    else return 1.0;
}


int main()
{
    for(int i = 0; i < 10;i++)
    {
        std::cout << "2^" << i << " = " << power(2,i) << std::endl;
    }
    return 0;
}
```

The power algorithm was able to correctly give the powers of two

Output

```
2^0 = 1
2^1 = 2
2^2 = 4
2^3 = 8
2^4 = 16
2^5 = 32
2^6 = 64
2^7 = 128
2^8 = 256
2^9 = 512
```

Question 3

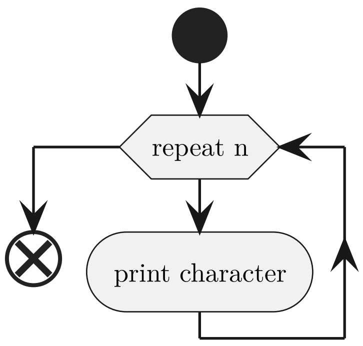

Figure 1: diagram

program.cpp

```cpp
#include <iostream>

int main()
{
    int n = 10;
    char character = '*';
    for(int i = 0; i < n; i++)
    {
        std::cout << character;
    }
    return 0;
}
```

Alternative

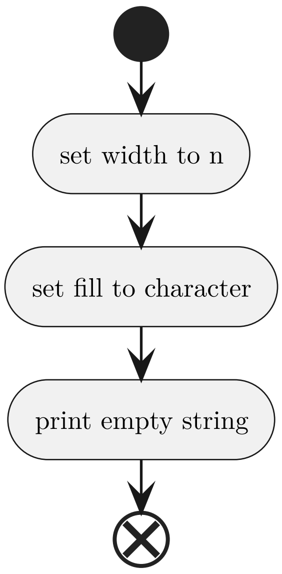

Figure 2: diagram

program.cpp

```cpp
#include <iostream>
#include <iomanip>

int main()
{
    int n = 10;
    char character = '*';
    std::cout << std::setw(n);
    std::cout << std::setfill(character);
    std::cout << "";
    return 0;
}
```

Conclusion

This is the conclusion of question 3

Question 4

program.py

```python
import matplotlib.pyplot as plt
import numpy as np


def func(x):
    return np.pow(x, 2)


X = np.linspace(0, 10, 100)
plt.plot(X, func(X))
plt.show()
```

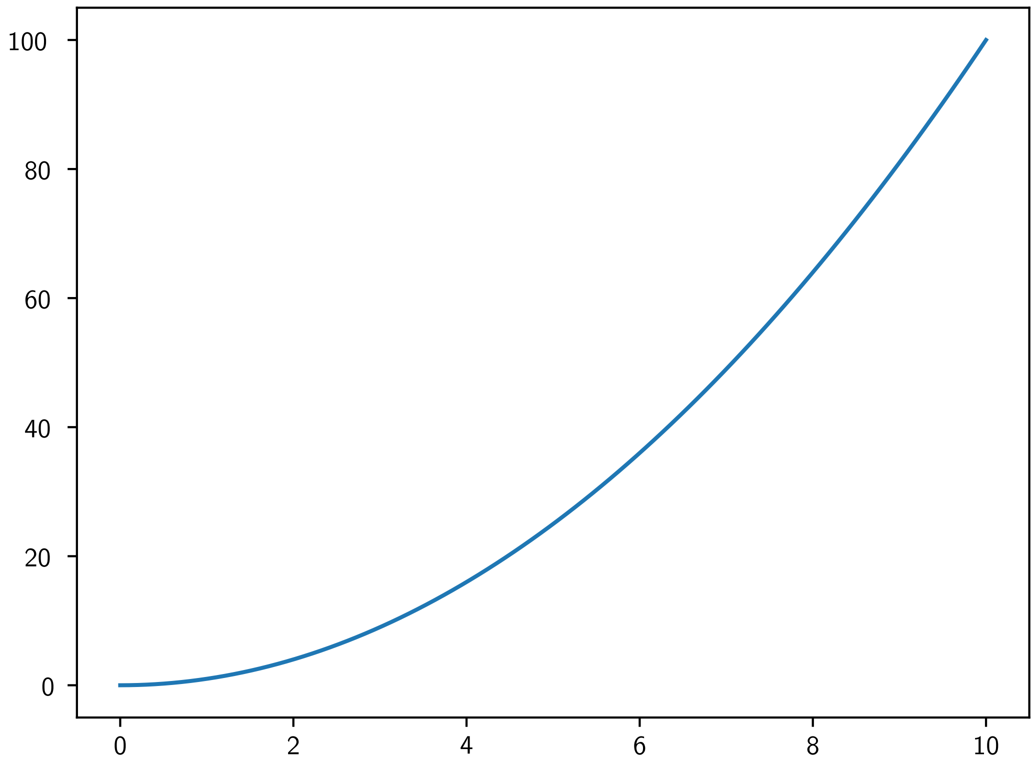

Figure 3: Figure 1