



به نام خدا



دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

دانشکده مهندسی مکانیک

رباتیک و مکاترونیک

تمرین شماره صفر

نام و نام خانوادگی	محمد منتظری
شماره دانشجویی	۸۱۰۶۹۹۲۶۹
تاریخ ارسال گزارش	۱۴۰۲/۱۲/۰۶

فهرست گزارش سوالات

- سوال ۱ - عملیات محاسباتی ۳
- سوال ۲ - عملگرهای شرطی ۵
- سوال ۳ - حلقه‌ها ۷
- سوال ۴ - توابع ۹
- سوال ۵ - نمایش‌سازی ۱۱

فهرست تصاویر

- شکل ۱ : نمودار دو موج سینوسی دارای نویز (با python) ۱۲
- شکل ۲: نمودار دو موج سینوسی دارای نویز (با C++) ۱۴

سوال ۱ - عملیات محاسباتی

در این سوال، به تعریف دو متغیر صحیح و اعشاری پرداخته و ۵ عملیات ریاضی جمع، تفریق، ضرب، تقسیم، و عملگر نمایی را روی آنها اعمال کرده و نتایج را با دو رقم اعشار با عنوان مناسب چاپ می‌کنیم.

کد پایتون:

```
# prob.1 - Arithmetic Operations
print("\nprob.1 - Arithmetic Operations")
x = 10
y = 4.12

Addition = x+y
Subtraction = x-y
Multiplication = x*y
Division = x/y
Exponentiation = np.exp(x)

print(f'Addition={Addition:.2f}, Subtraction={Subtraction},
Multiplication={Multiplication}, Division={Division:.2f},
Exponentiation={Exponentiation:.2f}')
```

نتایج:

```
prob.1 - Arithmetic Operations
Addition=14.12,      Subtraction=5.88,      Multiplication=41.2,      Division=2.43,
Exponentiation=22026.47
```

کد سی پلاس پلاس:

```
int prob1()
{
    int x = 10;
    float y = 4.12;

    float Addition = x + y;
    float Subtraction = x - y;
    float Multiplication = x * y;
    float Division = x / y;
    float Exponentiation = exp(x);
}
```

```
cout << "Addition=" << Addition;  
cout << "\nSubtraction=" << Subtraction;  
cout << "\nMultiplication=" << Multiplication;  
cout << "\nDivision=" << Division;  
cout << "\nExponentiation=" << Exponentiation;  
cout << endl;  
return 0;  
}
```

نتایج:

```
prob1 - Arithmetic Operations  
Addition=14.12  
Subtraction=5.88  
Multiplication=41.2  
Division=2.42718  
Exponentiation=22026.5
```

سوال ۲ - عملگرهای شرطی

در این سوال، عملگر شرطی if و else if (elif) و else را پیاده سازی کردیم تا با ورود سن کاربر به عنوان ورودی برنامه، دسته‌بندی سن او را طبق کلاس بندی زیر نمایش دهد:

1. Minor (Less than 18)
2. Adult (Between 18 and 65)
3. Senior (Greater than 65)

کد پایتون:

```
# prob.2 - Conditionals
print("\nprob.2 - Conditionals")
age = int(input("Please enter your age:\t"))
if age < 18:
    print("you're a Minor")
elif 18 <= age <= 65:
    print("you're an Adult")
elif 65 < age:
    print("you're a Senior")
else:
    print("invalid input")
```

نتایج:

```
prob.2 - Conditionals
Please enter your age: 22
you're an Adult
```

کد سی پلاس پلاس:

```
int prob2()
{
    int age;
    cout << "Please enter your age:\t";
    cin >> age;
    if (age < 18)
    {
        cout << "you're a Minor";
    }
    else if ((18 <= age) && (age <= 65))
    {
```

```
        cout << "you're an Adult";
    }
    else if (65 < age)
    {
        cout << "you're a Senior";
    }
    else
    {
        cout << "invalid input";
    }
    cout << endl;
    return 0;
}
```

نتائج:

```
prob2 - Conditionals
Please enter your age: 22
you're an Adult
```

سوال ۳ - حلقه‌ها

در این سوال، از حلقه for استفاده می‌کنیم تا دنباله فیبوناچی را تا تعداد جملات مشخصی چاپ کنیم. برای این منظور، به عنوان ورودی برنامه، متغیر n را برابر با ۱۲ جمله تعریف می‌کنیم. در [ریاضیات](#)، سری فیبوناچی (به [انگلیسی](#) Fibonacci number) به دنباله‌ای از [اعداد](#) می‌گویند که به صورت زیر تعریف می‌شود:

$$F(n) := \begin{cases} 1 & \text{if } n = 1; \\ F(n-1) + F(n-2) & \text{if } n > 1. \end{cases}$$

غیر از دو عدد اول، اعداد بعدی از جمع دو عدد قبلی خود به دست می‌آیند. اولین اعداد این سری عبارت‌اند از: ۰، ۱، ۱، ۲، ۳، ۵، ۸، ۱۳، ۲۱، ۳۴، ۵۵، ۸۹، ۱۴۴، ۲۳۳، ۳۷۷، ۶۱۰، ۹۸۷، ۱۵۹۷، ۲۵۸۴، ۴۱۸۱، ۶۷۶۵، ۱۰۹۴۶، ۱۷۷۱۱، ۲۸۶۵۷، ۴۶۳۶۸

این اعداد به نام [لئوناردو فیبوناچی](#)، ریاضی‌دان ایتالیایی قرن سیزدهم میلادی، نام‌گذاری شده‌است^۱.

کد پایتون:

```
# prob.3 - Loops
print("\nprob.3 - Loops")
n = 12
fib = [0, 1]
for i in range(n-2):
    new = fib[-1] + fib[-2]
    fib.append(new)
print(fib)
```

نتایج:

```
prob.3 - Loops
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
```

^۱ https://en.wikipedia.org/wiki/Fibonacci_number اعداد فیبوناچی - ویکی‌پدیا، دانشنامه آزاد

کد سی پلاس پلاس:

```
int prob3()
{
    int n = 12;
    int fib[n];
    int first = 0, second = 1;
    for (int i = 0; i < n; i++)
    {
        fib[i] = first;
        fib[i + 1] = second;
        int New = first + second;
        first = second;
        second = New;
    }

    for (int i = 0; i < n; i++)
    {
        cout << fib[i];
        if (i != n - 1)
        {
            cout << ", ";
        }
    }
    cout << endl;
    return 0;
}
```

نتایج:

prob3 - Loops

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89

سوال ۴ - توابع

در این سوال، یک تابع تحت عنوان prob4 تعریف می‌کنیم که به عنوان آرگومان، سه پارامتر n، d، و word را دریافت می‌کند. سپس کلمه پاس داده شده به تابع تحت عنوان پارامتر word را توسط یک حلقه به تعداد n بار چاپ می‌کنیم؛ درحالی که بین هر بار چاپ، به کمک کتابخانه‌های موجود، یک مکث d ثانیه‌ای برقرار می‌کنیم. در نهایت با فراخوانی تابع و پاس دادن کلمه Test با ۵ بار تکرار و ۱ ثانیه delay، نتایج را مشاهده می‌کنیم.

کد پایتون:

```
# prob.4 - Functions
print("\nprob.4 - Functions")

def prob4(n, d, word):
    for i in range(n):
        if i != n-1:
            end = ' - '
        else:
            end = "\n"
        print(word, end=end, flush=True)
        time.sleep(d)

prob4(5, 1, "Test")
```

نتایج:

```
prob.4 - Functions
Test - Test - Test - Test - Test
```

کد سی پلاس پلاس:

```
int prob4(int n, float d, string word)
{
    for (int i = 0; i < n; i++)
    {
        cout << word;
        if (i != n - 1)
        {
            cout << " - ";
            Sleep(1000*d);
        }
    }
}
```

```
cout << endl;  
return 0;  
}
```

نتایج:

prob4 - Functions
Test - Test - Test - Test – Test

سوال ۵ - نمایش سازی

در این سوال، قصد داریم دو موج سینوسی ایجاد کنیم و بعد از اضافه کردن نویز به آنها، توسط مجموعه‌ای از اعداد کوچک تصادفی (random) به عنوان نویز، آنها را در یک figure نمایش می‌دهیم. علاوه بر این، جزئیاتی همچون راهنمای نمودار (legend)، عنوان نمودار، عنوان محورها و غیره را به شکل اضافه می‌کنیم.

کد پایتون:

```
# prob.5 - Visualization
print("\nprob.5 - Visualization")
x = np.linspace(0, 10, 1000)
noise_1 = []
noise_2 = []
for i in range(1000):
    noise_1.append(random.uniform(0, 0.05))
    noise_2.append(random.uniform(0, 0.2))

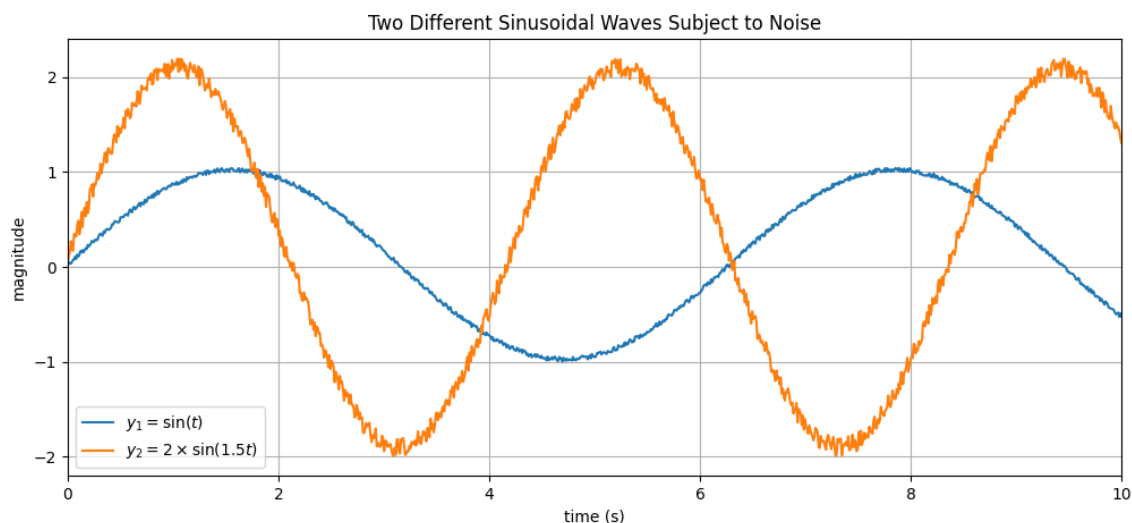
noise_1 = np.array(noise_1)
noise_2 = np.array(noise_2)

wave_1 = np.sin(x) + noise_1
wave_2 = 2*np.sin(1.5*x) + noise_2

plt.figure(1, (12, 5))
plt.plot(x, wave_1, label=r"$y_1 = \mathrm{sin}(t)$")
plt.plot(x, wave_2, label=r"$y_2 = 2\times\mathrm{sin}(1.5t)$")
plt.title("Two Different Sinusoidal Waves Subject to Noise")
plt.xlabel("time (s)")
plt.ylabel("magnitude")
plt.xlim([0, 10])
plt.legend()
plt.grid()
plt.show()
```

نتایج:

prob.5 – Visualization



شکل ۱: نمودار دو موج سینوسی دارای نویز (با python)

کد سی پلاس پلاس:

برای پیاده‌سازی این برنامه در C++ از ترفند خاصی استفاده کردیم. با تحقیق در نت و یوتیوب^۱، به این نتیجه رسیدیم که بهترین روش برای plot کردن نمودار در Cpp، استفاده از کتابخانه کارآمد matplotlib است که با اندکی چالش همراه است. برای انجام این کار، باید به نوعی کتابخانه مذکور را در Cpp پیاده‌سازی کنیم. برای اینکار، فایل matplotlibcpp.h را از گیت‌هاب دانلود کرده و در کنار source code ضمیمه می‌کنیم. سپس با وارد کردن آن در برنامه توسط خط:

```
#include "matplotlibcpp.h"
```

و تعریف namespace مربوطه در خط:

```
namespace plt = matplotlibcpp;
```

آماده می‌شویم که از دستوراتی مشابه با python در کد استفاده کنیم تا نتایج مشابهی حاصل کنیم. این برنامه از آنجایی که از matplotlibcpp.h استفاده می‌کند، قابل ران شدن به طور معمول نیست؛ بلکه باید در ترمینال دستوراتی وارد کرد که آدرس interpreter نصب شده برای python را به همراه آدرس محل نصب کتابخانه‌هایی همچون numpy و غیره به سیستم‌عامل و g++ compiler معرفی می‌کند. به همین خاطر، فایل جداگانه‌ای برای این سوال قرار داده شده است و برای جلوگیری از مواجهه با خطای compile error، این بخش از پروژه به صورت کامنت در فایل اصلی نیز قرار داده شده تا انسجام مطالب حفظ شود.

کد مذکور به شکل زیر است:

^۱ [Plotting in C++ using MATPLOTLIB \(g++ and cmake\) - YouTube](#)

```

#include <iostream>
#include <vector>
#include <cmath>
#include <stdlib.h>
#include <time.h>
#include "matplotlibcpp.h"

using namespace std;
namespace plt = matplotlibcpp;

int prob5()
{
    srand(time(0));
    vector<float> t;
    vector<float> wave_1, wave_2;
    for (int i = 0; i < 1000; i++)
    {
        t.push_back(i / 100.0);
        wave_1.push_back(sin(t[i]) + float(rand()) / float(RAND_MAX) *
0.1);
        wave_2.push_back(2 * sin(1.5 * t[i]) + float(rand()) /
float(RAND_MAX) * 0.35);
    }

    plt::figure_size(1300, 400);
    plt::plot(t, wave_1, "r", {"label", "y_1 = sin(t)"});
    plt::plot(t, wave_2, "b", {"label", "y_2 = 2*sin(1.5t)"});

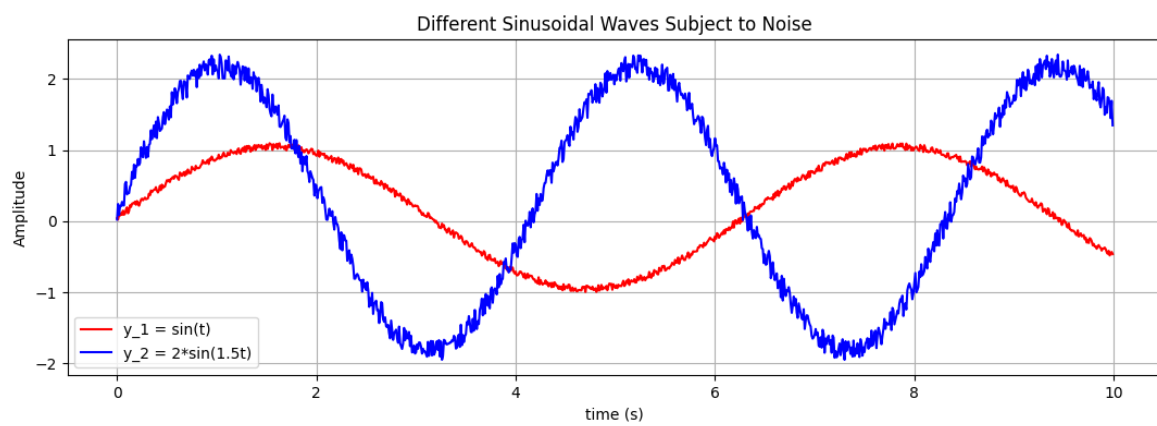
    plt::title("Different Sinusoidal Waves Subject to Noise");
    plt::xlabel("time (s)");
    plt::ylabel("Amplitude");
    plt::grid(true);
    plt::legend();
    plt::show();
    return 0;
}

int main()
{
    cout << "\nprob5 - Visualization";
    prob5();
    return 0;
}

```

نتایج:

prob5 – Visualization



شکل ۲: نمودار دو موج سینوسی دارای نویز (با C++)