

# 1 Learnig basic C programing (1<sup>st</sup> day)

## 1.1 Purpose

今回の実験ではC言語プログラミングの基本的な操作を理解する.

## 1.2 Procedure

まず Windows OS のノートパソコンを用意し”Visual C++ 2010 Express”を起動する. その後与えられた 1.1 から 1.3 の課題に取り組む.

### Task 1.1

整数 1 から 12234 までの総和を表示せよ.

### Task 1.2

1 から 100 までの数を配列に格納したのち奇数の総和を表示せよ.

### Task 1.3

1000 個の点をとることにより正弦波 (振幅=1, 位相=0)5 波長分の値を text ファイルに出力しその出力結果をエクセルを用いて表示せよ.

## 1.3 Result

実験の結果まず,task 1.1 では 74841495 の値が得られた.

また task 1.2 では 25000 の値が得られた.

さらに task 1.3 では図 1 のようなグラフがエクセルにより得られた.

## 1.4 Discussion

今回の課題 task1.1 において 1 から 12234 までの総和を for 文を用いて計算を行なったが総

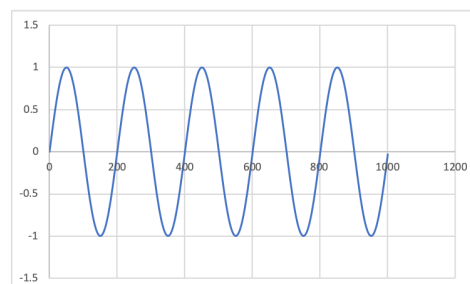


図 1: 振幅=1, 位相=0 の正弦波

和の公式から以下のようにも書き換えられることが予想できる.

$$\sum_{k=1}^n k = \frac{n(n+1)}{2}$$

この数式を用いて総和を求める場合の計算時間は  $O(1)$  で計算できる一方で for 文を用いて総和を求める場合は  $O(N)$  の時間がかかることが予想できる. またこの数式に  $n=12234$  を代入すると確かに 74841495 の値を得ることができるので今回の結果は正しいと考えられる.

また, task1.2 においても前問同様に総和の公式を用いると

$$\begin{aligned} \sum_{k=1}^n 2k-1 &= 2 \frac{n(n+1)}{2} - n \\ &= n^2 \end{aligned} \quad (1)$$

この式において  $n=50$  を代入すると 2500 の値が得られるのでプログラムによる計算結果は正しいと考えられる. また, 今回も同様に総和公式を使った場合の最悪時間計算量は  $O(N)$  で抑えられることができる.

## 1.5 Appendix

今回の課題で使ったソースコードを以下に示す.

ソースコード 1: task1.1

---

```
1 #include <iostream>
2 #include <math.h>
3 int main(){
4     int sum = 0;
5     for(int i=1; i<=12234;i++)
6     {
7         sum += i;
8     }
9     cout << sum << endl;
10 }
```

---

ソースコード 2: task1.2

---

```
1 #include <iostream>
2 #include <math.h>
3 int main(){
4     int sum = 0, a[100];
5     for(int i = 0; i < 100; i++){
6         a[i] = i + 1;
7         if(i%2 == 0){
8             sum += a[i];
9         }
10    }
11    cout << sum << endl;
12 }
```

---

ソースコード 3: task1.3

---

```
1 #include <iostream>
2 #include <stdio.h>
3 #include <math.h>
4 #define PI 3.141592653
5 int main(){
6     double a[1000];
7     FILE *fp;
8     fp = fopen("test.txt","w");
9     for(int i = 0; i < 1000; i++)
10    {
11        a[i] = sin(PI*i/100);
12    }
13    for(int i = 0; i < 1000; i++)
14    {
15        fprintf(fp,"%f\n",a[i]);
16    }
17    fclose(fp);
18 }
```

---