第8回 C++プログラミング実験II 実験課題

課題実施日:2022 年 5 月 30 日. 締切 6 月 2 日(木) 24:00

標準入力によって得られたデータの平均，分散，中央値を計算するプログラム(ex08-1-main.cpp)を完成させよ. ただし，データの個数が奇数の場合，中央値はデータを整列させたときの真ん中の値(例:1，2，3の中央値は2)である. また，データの個数が偶数の場合，中央値はデータの中央付近に位置する2つの値の平均(例:1，2，3，4の中央値は(2+3)/2=2.5)である. 実行例の通り，データは重複ありデータであることに注意する.

```cpp
//ex08-1-main.cpp
#include <iostream>
#include <iterator>
#include <set>

double med(std::multiset<double> ms)
{
//データの中央値を計算する
}

double mean(std::multiset<double> ms)
{
//データの平均を計算する
}

double var(std::multiset<double> ms)
{
//データの分散を計算する
}

int main()
{
    std::multiset<double> ms;
    std::istream_iterator<int> it{std::cin};
    std::istream_iterator<int> eos;

    for (; it != eos; ++it)
    {
        //ここを埋める(データの格納を行う.「>>演算子」は使用しないこと)
    }
```

```cpp
//ex08-1-main.cpp 続き
    std::cout << "sort_data: " << "¥n";
    for (const double e : ms)
    {
        std::cout << e << "¥n";
    }

    std::cout << "median = " << med(ms) << "¥n";
    std::cout << "mean = " << mean(ms) << "¥n";
    std::cout << "variance = " << var(ms) << "¥n";

    return 0;
}
```

//実行例1（下線部はキーボード入力）

$ g++ -std=c++17 ex08-1-main.cpp -o ex08-1

$./ex08-1

5

4

3

2

1

Ctrl+d

sort_data:

1

2

3

4

5

median = 3

mean = 3

variance = 2

//実行例2

$ g++ -std=c++17 ex08-1-main.cpp -o ex08-1

$./ex08-1

1

2

3

4

5

6

Ctrl+d

sort_data:

1

2

3

4

5

6

median = 3.5

mean = 3.5

variance = 2.91667

実験課題 8-2

第 5 回目の授業スライドの random.hpp を用いて，文字の出現頻度を描画するプログラムを完成させよ(ex08-2-main.cpp).

```cpp
//ex08-2-main.cpp
#include <iostream>
#include <map>
#include <list>
#include "random.hpp"

void count(const std::list<char>&data, std::map<char, std::string>& histogram){
    // 1
}
void print(const std::map<char, std::string>& histogram){
    // 2
}
int main(){
    const size_t N{100};
    ExpDist e(0.5);
    std::list<char> data;
    std::map<char, std::string> histogram;

    for(size_t i{0}; i<N; i++) data.insert(data.begin(), e.get()+97);

    count(data, histogram);
    print(histogram);

    return 0;
}
```

以下は作成する関数の説明である.

1. count

   第一引数の乱数 data を用いて第二引数の histogram に頻度の情報を格納する.

   例えば，a が 3 回出現したときのヒストグラムは histogram['a']="---"となる.

2. print

   実行例のような，a,b,c,d,e のみを出力するヒストグラムを表示する.

```
// 実行例
./a.out
a ----------------------------------
b ------------------------
c -------------
d ------
e ------
```

```
// 実行例
./a.out
a ----------------------------------
b ------------------------
c -------------
d ---------
e ------
```

```
// 実行例
./a.out
a ----------------------------------
b ------------------------
c ------------
d ---------
e ------
```

```
// 実行例
./a.out
a ----------------------------------
b --------------------
c ----------------
d --------
e ----------
```

実験課題 8-3

　ストリームイテレータを用いて、入力された数値を非負数と負の数を２つの配列に分類して、さらにそれぞれの数値と要素数を出力するプログラムを作成せよ。必ずスケルトンファイルに記載の指示に従うこと。またすでにスケルトンファイルに記載済みのコードは変更してはならない。

注： std::copy()はイテレータを用いて指定された範囲をコピーする関数。

ex 8-3_skl.cpp

```
#include <iostream>
#include <iterator>
#include <vector>


int main(int argc, const char * argv[]) {

    //vectorの定義
    std::vector<int> posi, nega;

    //(1)入力イテレータを使って入力処理


    //(2)非負数と負の数を２つの配列に振り分け
    for (     ){

    }

    //(3)出力イテレータを使って配列に格納した非負数と要素数の出力（x.sizeおよびサイズ関数の使用は禁止）
    if( /*イテレータを用いた要素数判定*/ ){
    std::copy(/*先頭要素のイテレータ*/,/*末尾要素のイテレータ*/, /*出力イテレータ(オブジェクトなし)*/));
    std::cout<<"Num of Plus = "<</*イテレータを用いた要素数の取得*/<<std::endl;
    }
    else
    std::cout<<"Num of Plus = 0"<<std::endl;

    //(4)出力イテレータを使って配列に格納した負の数と要素数の出力（x.sizeおよびサイズ関数の使用は禁止）
    if( /*イテレータを用いた要素数判定*/ ){
    std::copy(/*先頭要素のイテレータ*/,/*末尾要素のイテレータ*/, /*出力イテレータ(オブジェクトなし)*/);
    std::cout<<"Num of Minus = "<</*イテレータを用いた要素数の取得*/<<std::endl;
    }
    else
    std::cout<<"Num of Minus = 0"<<std::en
```

8-3 実行例

```
$./ex08-3
-1 0 1 -2 -3 -5 697 -3 9 68 78 -1 [Ctrl+D](Xcode：break)
0 1 697 9 68 78 Num of Plus = 6
-1 -2 -3 -5 -3 -1 Num of Minus = 6
```

```
$./ex08-3
0 1 2 3 4 [Ctrl+D](Xcode：break)
0 1 2 3 4 Num of Plus = 5
Num of Minus = 0
```

```
$./ex08-3
-3 -6 -80 5 -100 2 -32 7 [Ctrl+D](Xcode：break)
5 2 7 Num of Plus = 3
-3 -6 -80 -100 -32 Num of Minus = 5
```

```
$./ex08-3
-1 -3 -70 -100 [Ctrl+D](Xcode：break)
Num of Plus = 0
-1 -3 -70 -100 Num of Minus = 4
```