

問題1 (提出期限 本日 15:00) string

以下は、好みの果物（リンゴかオレンジ）の必要数を聞くプログラムである。プログラム実行例に合うように空欄(1)–(6)を適切に埋めてプログラムを完成させよ。空欄以外の部分は変更しないこと。空欄は1行か2行。（プログラム全体を ex3-1.cpp として提出すること）

```
// ex3-1.cpp
#include <iostream>
int main()
{
    (1) string 変数 apl と org を適切に初期化して宣言する

    std::string answer{};
    int kosu{};
    // 複数の string を, <<で別々に表示

    std::cout<<"Which do you like " << apl << " or " << org << "? ";
    std::cin>> answer;
    // 複数の string を1つの string へ連結してから, 表示

    if( (2) )      std::cout<<"You like " + answer + ". I will prepare red ones!\n";
    else if( (3) )  std::cout<<"You like " + answer + ". I have juicy ones!\n";
    else {
        std::cout<<"Please select either " + apl + " or " + org + " .\n";
        return 0; // main 関数から戻る, すなわち, この時点でプログラムは終了
    }

    std::string st;
    (4) 実行例の3行目の出力になるように, 適切な値を st に代入する

    std::cout<<st;
    std::cin>> kosu;

    if( kosu == 1 )      std::cout<< _____ (5)
    else if( kosu > 1 )  std::cout<< (6) 表示に string ではなく int の kosu が混じっていることに注意
    else
        std::cout<<"Please input a positive number!\n";
    return 0;
}
```

実行例（下線灰色はキーボード入力跡、プログラムからの出力ではないことに注意！）

プログラムの空欄(5)(6)は、黄色で示すように単語単数か複数も考慮のこと

```
% ./ex3-1
Which do you like apple or orange? apl
Please select either apple or orange.

% ./ex3-1
Which do you like apple or orange? apple
You like apple. I will prepare red ones!
How many apples do you want? 0
Please input a positive number!

% ./ex3-1
Which do you like apple or orange? apple
You like apple. I will prepare red ones!
How many apples do you want? 2
You want 2 apples.
```

```
% ./ex3-1
Which do you like apple or orange? orange
You like orange. I have juicy ones!
How many oranges do you want? 1
You want only one orange.

% ./ex3-1
Which do you like apple or orange? orange
You like orange. I have juicy ones!
How many oranges do you want? 3
You want 3 oranges.
```

問題2 (提出期限 10月10日 23:59) double の vector

以下のプログラムは、二次元座標値を vector 変数に設定したり、ベクトルの距離（2点間のユークリッド距離）を求めたり、2つの二次元ベクトルのベクトル和を求めるプログラムである。空欄(0)-(5)をプログラムの実行例に合うように適切に埋めてプログラムを完成させよ。（プログラム全体を ex3-2.cpp として提出すること）注意：既存のコードに加えることはできるが、変更や削除をしてはならない。

```
// ex3-2.cpp
#include <iostream>
// (0) 必要なヘッダーファイルを定義する
int main()
{
    std::vector<double> p1{1.0,1.0}; // 要素数2の初期化したvector変数 p1 p1(1.0, 1.0)

    // サイズを指定して宣言したvectorに、キーボード入力を直接行う方法
    std::vector<double> p2(2); // 要素数2のvector変数 p2 サイズのみ指定、中身は0
    (1) p2の値にキーボードからの入力値を設定
    // 空のvectorを用意して、キーボード入力した値を、vectorに挿入(push_back)する方法
    std::vector<double> p3; // 要素数0のvector変数
    (2) p3の値にキーボードからの入力値を設定
    // p1 p2 p3座標表示
    (3) 実行例のフォーマットで、上記の3つの点の座標を表示

    // p1とp2の距離を計算して表示
    (4) double変数distanceに p1とp2の距離を計算する（第1回実験演習の問題3解答例を参考に）
    std::cout<< "The distance of p1 and p2 is " << distance <<"\n";

    // p2とp3を二次元ベクトルとした場合のベクトルの和p4を計算して表示
    std::vector<double> p4(2); // 要素数2のvector 値は0
    (5) ベクトルの和 p4 の計算と表示
    return 0;
}
```

実行例（下線灰色はキーボード入力の跡，プログラムからの出力ではないことに注意！）

```
% ./ex3-2
Input 2D point p2 ( x and y ) ! 2.4
Input 2D point p3 ( x and y ) ! 5.2 6.8
p1:(1, 1)
p2:(2, 4)
p3:(5.2, 6.8)
The distance of p1 and p2 is 3.16228
p4:(7.2, 10.8)
```

```
% ./ex3-2
Input 2D point p2 ( x and y ) ! 1.4
Input 2D point p3 ( x and y ) ! -2.4 -3.3
p1:(1, 1)
p2:(1, 4)
p3:(-2.4, -3.3)
The distance of p1 and p2 is 3
p4:(-1.4, 0.7)
```

問題 3（提出期限 10 月 10 日 23:59） string の vector と int の vector

3 商品（今回はコーラとポテトとハンバーガー）のみを売る商店で、価格表を表示した後、それぞれの注文個数を客に入力させ、注文個数の確認表示と最終の支払い額を表示するプログラムを作成せよ。

以下の 4 つの条件を満たす適切な vector を作成して用いよ。入出力表示は実行例と同様にすること。入力される注文個数は 0 以上の整数が必ず入力 されるものと仮定してよい。

1. 商品の名称の vector と、商品の価格の vector を用意し、それぞれ初期化して用いること。
2. 客が入力した 3 つの商品の注文個数を格納する vector を用意してこれに代入すること。
3. 注文個数の確認表示と合計料金の計算は、上記の vector を用いて行うこと。料金には、10% の消費税がかかるので、これを含めて最終支払額を表示すること。
4. 1 で定義した 2 つの vector の初期値を書き換えるだけで、扱う 3 つの商品の名称と価格を変更できるようにプログラムを作成すること。

実行例（下線灰色はキーボード入力の跡，プログラムからの出力ではないことに注意！）

```
% ./ex3-3
Welcome to our shop!
Please input required number for each item!
----- Price List -----
cola            300 yen
potato          350 yen
burger          580 yen
-----

Input num of cola! 2
Input num of potato! 1
Input num of burger! 2

Thank you! Items you ordered are
cola 2
potato 1
burger 2
Total price (including 10% tax) is 2321 yen
```

```
% ./ex3-3
Welcome to our shop!
Please input required number for each item!
----- Price List -----
cola            300 yen
potato          350 yen
burger          580 yen
-----

Input num of cola! 1
Input num of potato! 0
Input num of burger! 2

Thank you! Items you ordered are
cola 1
potato 0
burger 2
Total price (including 10% tax) is 1606 yen
```

問題3－3 条件4の補足説明

前ページの条件4

1で定義した2つのvectorの初期値を書き換えるだけで、扱う3つの商品の名称と価格を変更できるようにプログラムを作成すること。

この意味は、

作成したプログラムで、商品の名称のvectorと、商品の価格のvectorの宣言文の初期値（商品名と価格）を変更するだけで、他の部分の変更なしに、ハンバーガーショップから、例えば、以下の様にアイスクリームショップに変更できるようなプログラムにしてほしいということである。

以下は、ex3-3.cppのうちvector宣言文2行を変更したプログラムex3-3b.cppの実行例


ハンバーガーショップ

扱う商品を変更して、 アイスクリームショップへ

```
% ./ex3-3
Welcome to our shop!
Please input required number for each item!
----- Price List -----
cola          300 yen
potato        350 yen
burger        580 yen
-----

Input num of cola! 2
Input num of potato! 1
Input num of burger! 2

Thank you! Items you ordered are
cola 2
potato 1
burger 2
Total price (including 10% tax) is 2321 yen
```



```
% ./ex3-3b
Welcome to our shop!
Please input required number for each item!
----- Price List -----
Vanilla Ice cream 300 yen
Chocolate Ice cream 350 yen
Strawberry Ice cream 400 yen
-----

Input num of Vanilla Ice cream! 2
Input num of Chocolate Ice cream! 0
Input num of Strawberry Ice cream! 1

Thank you! Items you ordered are
Vanilla Ice cream 2
Chocolate Ice cream 0
Strawberry Ice cream 1
Total price (including 10% tax) is 1100 yen
```

ex3-3b.cppを作成して、提出する必要はない。

今回は、扱う商品の種類数は3種に固定である。

これが10商品あったら、プログラムは長く、書くのも面倒である...

次回の講義で、くりかえし文を学ぶと、商品の種類数も容易に変更しやすく、短いプログラムになる。