

ご参加の前に

スパコン体験塾は、C 言語や C++ を使ったプログラミングについて基本的な知識を持つ方を対象としています (C++ については C++11 レベルを想定)。以下の準備課題にチャレンジし、自分の知識を確認してみましょう。

※チャレンジをするときは、本やインターネット上で調べた情報を使って構いません。

下記のサイトでは、自分のパソコンに C 言語の環境がインストールされていなくても、オンライン上で C 言語のプログラムを書いてコンパイルし、実行することができます。準備課題へのチャレンジに役立ててください。

tutorialspoint https://www.tutorialspoint.com/compile_c_online.php

programiz <https://www.programiz.com/c-programming/online-compiler/>

paiza.io <https://paiza.io/ja/projects/new?language=c>

チャレンジ 1

文字列の入力と出力

出力例を参考にして、下線部分を埋めてプログラムを完成させよう。

注意：この課題で paiza.io を使用する場合は、実行前に「入力」に文字列を入力してください。

<出力例> (100 をコマンドライン引数として入力した場合)

```
./a.out 100
Your input is 100
```

```
1 // プログラム (未完成)
2 // おまけ: 画面入力から読み取った値を n に代入するよう書き換えてみよう。
3 #include <cstdlib>
4 #include <iostream>
5 #include <string>
6
7 int main(int argc, char *argv[])
8 {
9     if (argc != 2) {
10        std::cout << "[Usage] ./a.out (arg1)" << std::endl;
11        std::cout << "[Usage] (arg1): integer" << std::endl;
12        return 0;
13    }
14    int n;
15    // 1 番目のコマンドライン引数を読み取り、整数に変換する。
16    n = std::atoi(____);
17    // string 型の変数 msg に出力例に合わせて文字列を記入する。
18    std::string msg = _____;
19    // C の printf 関数を使って、画面に文字列と入力した整数を表示する。
20    printf("%s %d\n", msg.c_str(), ____);
21    return 0;
22 }
```

チャレンジ 2

関数と配列、整数の出力

出力例を参考にして、下線部分を埋めてプログラムを完成させよう。ただし、関数 func で実行する計算 (アルゴリズム) は次のものとする。

- ① 配列 a の先頭要素 (a[0]) は変更しない。
- ② 配列 a の i 番目の要素は、配列 a の i-1 番目の要素と配列 b の i 番目の要素を加えたものとする (i ≥ 1)。

<出力例>

```
# va vb
0 5
2 2
2 0
7 5
```

```
1 // プログラム (未完成)
2 // おまけ: 配列ではなく、vector コンテナを使って書き換えてみよう。
3 #include <cstdlib>
4 #include <random>
5 #include <string>
6 #define NSIZE 4
7 void func (int n, _____)
8 { // アルゴリズムの実装。
9     for (____; i < n; ++i) {
10        a[i] = a[____] + b[i];
11    }
12 }
13
14 int main (int argc, char **argv)
15 {
16    const uint32_t iseed = 55;
17    std::mt19937 engine(iseed);
18    int va[NSIZE], vb[NSIZE];
19    std::uniform_int_distribution<int> distint(0,5);
20    for (int i = 0; i < NSIZE; ++i) {
21        va[i] = 0;
22        vb[i] = distint(engine);
23    }
24    // 配列 va と vb を関数を渡すように渡す。
25    func(NSIZE, va, vb);
26    printf("#%3s %3s\n", "va", "vb");
27    // 配列 a と b の要素を全て画面に書き出す。
28    for (int i = 0; i < NSIZE; ++i) {
29        printf("%_____ %\n", va[i], vb[i]);
30    }
31    return 0;
32 }
```

チャレンジ 3

簡単なシミュレーション、浮動小数点数の出力

出力例を参考にして、下線部分を埋めてプログラムを完成させよう。ただし、x および v の更新規則 (アルゴリズム) は次のものとする。

$x \leftarrow x + v * dt$, $v \leftarrow v + a * dt$ (古い x と v の値を使って、新しい x と v の値を計算する)

<出力例>

x について公式 $x = x_0 + v_0 t + gt^2/2$ の計算値と比較してみよう

```
$. /a.out
time      x      v
0.0    333.0    0.0
1.0    328.1   -9.8
```

```
1 // プログラム (未完成)
2 #include <cstdlib>
3 #define NTIME 500
4
5 int main (int argc, char **argv)
6 {
7     const double dt = 2.0e-3;
8     const double g = 9.8;
9     const double t0 = 0.0;
10    const double x0 = 333.0; // x の初期値 x0
11    const double v0 = 0.0; // v の初期値 v0
12
13    double t, x, v;
14    x = x0;
15    v = v0;
16    t = t0;
17    printf("%9s %9s %9s\n", "time", "x", "v");
18    printf("_____ \n", t, x, v);
19
20    for (int it = 1; it <= NTIME; ++it) {
21        double a = -1.0 * g;
22        x = _____ + v * dt; // x の値の更新
23        _____ = v + a * dt; // v の値の更新
24        t = t + dt;
25    }
26    printf("_____ \n", t, x, v);
27    return 0;
28 }
```