ご参加の前に

スパコン体験塾は、C 言語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。
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。

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＞（RIST をキーボードから入力した場合）

```
$ ./a.out
Enter your name (Within 16 characters, No space):
RIST
Hello. I am RIST. My ID is 001.
```

```
1  /* プログラム（未完成）*/
2  #include <stdio.h>
3  int main (int argc, char **argv)
4  {
5    char cbuff[16];
6    char str[256];
7    int id = 1;
8    printf("Enter your name (Within 16 characters, No space):\n");
9    /* 画面からの入力を読み取り cbuff に格納する */
10   scanf("%s", _____);
11   /* str に cbuff と id を書き込む */
12   _____(str, "Hello. I am %____. My ID is %____.\n", cbuff, id);
13   printf("%s\n", str);
14   return 0;
15 }
```

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＞

```
$ cat output
#a    b
  1    2
  3    2
  5    2
  5    0
```

```
1  /* プログラム（未完成）*/
2  #include <stdio.h>
3  #include <stdlib.h>
4  #define NSIZE 4
5  void func(const int n, int *a, int *b)
6  { /* アルゴリズムの実装 */
7    for (_____; i < n; ++i )
8      a[i] = a[____] + b[i];
9  }
10
11 int main (int argc, char **argv)
12 {
13   int a[NSIZE], b[NSIZE];
14   srand(55);
15   for (int i = 0; i < NSIZE; ++i ) {
16     a[i] = 1;
17     b[i] = rand() % 5;
18   }
19   /* 配列 a と b を関数で処理できるように渡す */
20   func (NSIZE,____,____);
21   printf("#%3s %3s\n", "a", "b");
22   /* 配列 a と b の要素を全て画面に書き出す */
23   for (int i=0; i<NSIZE; ++i)
24     printf(" %____ %____\n", a[i], b[i]);
25   return 0;
26 }
```

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 $x=x_0+v_0t+gt^2/2$ 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 <stdio.h>
3  #define NTIME 500
4
5  int main (int argc, char **argv)
6  {
7    double dt = 0.002;
8    double g = 9.8;
9    double x0 = 333.0; /* x の初期値 */
10   double v0 = 0.0;   /* v の初期値 */
11   double x, v, t;
12   x = x0;
13   v = v0;
14   t = 0.0;
15   printf("%9s %9s %9s\n", "time", "x", "v");
16   printf("%____ %____ %____\n",t, x, v);
17   for (int it = 1; it <= NTIME; ++it ) {
18     double a = -1.0 * g;
19     x =____+ v*dt; /* x の値の更新 */
20     v = v + a*dt;  /* v の値の更新 */
21     t = t + dt;
22   }
23   printf("%____ %____ %____\n",t, x, v);
24   return 0;
25 }
```