Introduction to Computer Science
Tutorial 8: Randomness, Arrays

1. Randomness

1.1 Fixing a mistake. This program is supposed to print a sequence of 10 pseudorandom numbers, but it does not work as intended. What is the issue and how can it be solved?

```
main() {
    int i;
    for (i = 0; I < 10; i++) {
        srand(time(NULL));
        printf("%d ", rand());
    }
}
```

1.2 Implementing a Password Generator

Write a program that prints n (n is provided by the user) passwords according to the following rules:

● each password is composed of 10 characters

● to generate a character, randomly choose a category, then randomly choose a character inside that category

● there are 3 categories of characters: a-z, A-Z, and 0-9. All three categories are mandatory (at least one character for each category exists in a valid password).

Here is an example of the desired output of one execution of the program (each execution of the program should yield a different output):

```
9E15GJ42f2
64iR47215p
oLaFXILG90
0H632gjMKW
S1MQE068v7
pAKSwNIgr7
8N3946D2mR
8314J12nDI
X63QG4b6jr
5f274U76E4
```

2. Output of Programs with Arrays

For each program, write down what would be its output when executed.

2.1

```
main(){
  int i;
  int a[5] = {10,20,30,40,50};
  a[1] = 33;
  a[4] = 66;
  for(i=0; i<5; i++)
    printf("%d ", a[i]);
}
```

2.2

```
main(){
  int i=1, j=2;
  int a[5] = {10,20,30,40,50};
  a[i+j] = 33;
  a[j*2] = 66;
  for(i=0; i<5; i++)
    printf("%d ", a[i]);
}
```

2.3

```
main(){
  int i;
  int a[5] = {10,20,30};
  a[4] = a[2];
  for(i=0; i<5; i++)
    printf("%d ", a[i]);
}
```

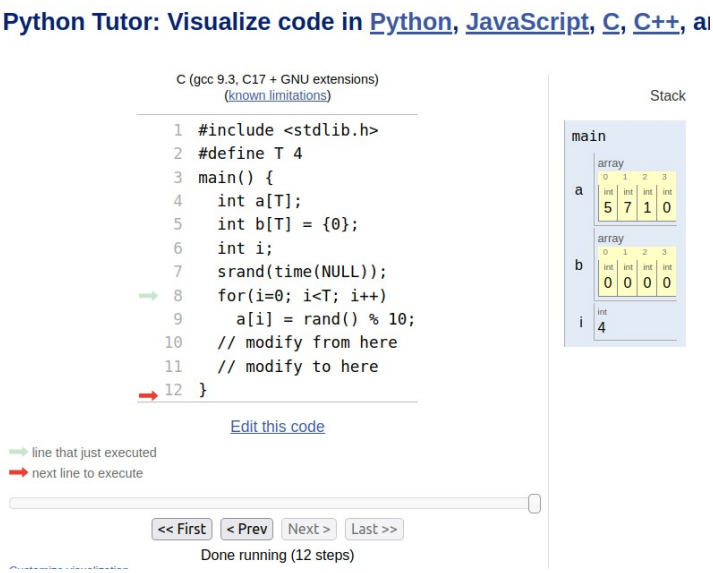
2.4

```
main(){
  int i;
  int a[5] = {10,7,13,8,15};
  int b[5] = {0};
  for(i=1; i<4; i++)
    b[i] = a[i+1];
  for(i=0; i<5; i++)
    printf("%d %d ",a[i], b[i]);
}
```

3. Completing and Visualizing Programs with Arrays

You are given programs that are *almost* complete. You need to add statements in the parts indicated by comments, to fulfill each corresponding task.

It is recommended to copy the provided programs to **https://pythontutor.com/c.html** and fill the missing instructions, so as to visualize the execution of the programs:



You may need to add an #include to use the NULL constant, like:
`#include <stdlib.h>`

3.1
The following program declares two arrays a and b; b is initialized with zero's and a is filled with random values from 0 to 9.

In the indicated place, **add statements to the main() function so that the contents of array `a[]` is copied into array `b[]`**.

```
#define T 4

main() {
  int a[T];
  int b[T] = {0};
  int i;
  srand(time(NULL));
  for(i=0; i<T; i++)
    a[i] = rand() % 10;
  // modify from here

  // modify to here
}
```

3.2

The following program declares two arrays a and b, and fills them with random values. It also declares a third array, c, initialized with zero's.

Add statements to the `main()` function so that the array `c[]` receives, at each position `i`, the value `a[i] + b[i]`.

```
#define T 4

main() {
  int a[T];
  int b[T];
  int c[T] = {0};
  int i;
  srand(time(NULL));
  for(i=0; i<T; i++){
    a[i] = rand() % 10;
    b[i] = rand() % 10;
  }
  // modify from here

  // modify to here
}
```

3.3

The following program fills an array a with random values from 1 to 5.

Modify the `main()` function so it calculates the product of the elements of array `a[]`, into a variable called `product`.

```
#define T 4

main() {
  int a[T];
  int i;
             // declare a new variable here
  srand(time(NULL));
  for(i=0; i<T; i++)
    a[i] = 1 + rand() % 5;
  // modify from here

  // modify to here
}
```

3.4

The following program fills an array a with random values from 1 to 5.

Modify the `main()` function so it finds the maximum value of the array `a[]` and stores it into a variable named `max`. Be careful with what you use as the initial value of `max`.

```
#define T 4

main() {
  int a[T];
  int i;
             // declare a new variable here
  srand(time(NULL));
  for(i=0; i<T; i++)
    a[i] = 1 + rand() % 5;
  // modify from here

  // modify to here
}
```

3.5

Consider the program below.

Modify the `main()` function so that the contents of array `a[]` gets copied into array `b[]` *in reverse order*.

For instance, if `a[]` contains `{3,6,0,0,2}`, `b[]` should contain `{2,0,0,6,3}` after the statements you will add.

```
#define T 4

main() {
  int a[T];
  int b[T] = {0};
  int i;
  srand(time(NULL));
  for(i=0; i<T; i++)
    a[i] = rand() % 10;
  // modify from here

  // modify to here

}
```