Introduction to Computer Science - Winter 2024

Tutorial 10: Strings, Functions and Arrays

1. Programs Outputs

1.1

What is the output of this C program?

```
main(){
  int i;
  char c[] = "Nice";
  for(i=0; i<5; i++)
    printf("%c ",c[i]);
}
```

1.2

What is the output of this C program?

```
void go(int a[], int j){
  printf("%d ", a[j]);
}

main(){
  int i;
  int a[5] = {3,3,7,3,8};
  int b[5] = {2,2,4,2,2};
  for(i=0; i<5; i++)
    go(b,i);
  go(a,b[i-2]));
}
```

2. Functions Definitions

2.1

In a C file, define the following functions:

- `void printArr(const int a[], int n)`
  - prints an array of size `n` on a single line and prints a newline character
- `void fillRandom(int a[], int n)`
  - fills the array with random values from 0 to 99

  - the function assumes that the calling environment has already initialized the random number generator with `srand()`
- `void reverse(const int a[], int b[], int n)`
  - copies the contents of array `a` in reverse order into array `b` (`n` is the size of both arrays)

In the same program, use the following `main()` function to test your functions:

```
#include <stdlib.h>
#include <time.h>
#include <stdio.h>

#define S 10
main(){
  srand(time(NULL));
  int x[S];
  int y[S];
  fillRandom(x,S);
  reverse(x,y,S);
  printArr(x,S);
  printArr(y,S);
}
```

2.2

In a C program,define the following functions:

- `int isLower(char c)`
  - returns 1 if c is a lowercase letter, 0 otherwise
- `int isUpper(char c)`
  - returns 1 if c is an uppercase letter, 0 otherwise
- `char modChar(char c)`
  - when given a lowercase letter, should returns its uppercase counterpart, and vice versa
  - when given any other kind of character, returns the same character
  - the function must call `isLower()` and `isUpper()`
- `void modString(char str[])`
  - applies the function `modChar()` to the zero-terminated string `str`, character by character

Use the following `main()` function to test your program:

```
main(){
  char input[100];
  gets(input);
  modString(input);
  printf("%s\n", input);
}
```

Example of execution:

```
$ tcc -w -run strings.c
Hello World!
hELLO wORLD!
```

2.3

In a C program, define a function: `int search(const int a[], int n, int v)`

The function performs a linear search of the value `v` in the array `a` of size `n`.

That is, the function's return value should be:

- the smallest value `i` such that `v == a[i]`, if such `i` exists, otherwise
- the symbolic constant value `ERROR`.

`ERROR` should be defined as `-1`.

In the same program, define a `main()` function that declares and initializes an array, and does two calls to `search()`, to show both behaviours of the function.