Lab 12

Q1:
Q2:

Solve the following 23 questions.
Submit your solution in the text box by writing each question number and the corresponding answer on a separate line.
For example,
Q1: b
Q2: c
.
.
.
Q23: d

Answer the following three questions (from q1 to q7) about the following program (named read_numbers.c) assuming that the user input is: 1275FFsnowtoy239.7

```
//read_numbers.c
#include <stdio.h>

int main() {
   int a = 1, b = 2;
   int c = 3;
   float x = 4, y = 5;
   char str[5] = "hello";
   int z = scanf("%2d %3o %4x %4s %*c %2f %*2c %3f", &a, &b, &c, str, &x, &y);

   return 0;
}
```

Q 1:

What is the number that will be printed by the line: printf("%d\n", a);?

a) 1275
b) 12
c) 1
d) 1275FF

Q 2:

What is the number that will be printed by the line: printf("%d\n", b);?

a) 75
b) 75F
c) 61
d) 1275

Q 3:

What is the number that will be printed by the line: printf("%d\n", c);?

a) FF
b) 255
c) 75
d) 3

Q 4:

What is the number that will be printed by the line: printf("%.2f\n", x);?

e) 239.00
f) 4.00
g) 23.00
h) 4

Q 5:

What is the number that will be printed by the line: printf("%.2f\n", y);?

e) 39.70
f) 5.00
g) 239.00
h) 9.70

Q 6:

What is the number that will be printed by the line: printf("%s\n", str);?

e) “hello”
f) “snowtoy”
g) “snow”
h) “s”

Q 7:

What is the number that will be printed by the line: printf("%d\n", z);?

a) 8
b) 6
c) 4
d) 1

Q 8:

For the following two statements: char *str1 = “capitalZ”; and char str2[10] = “capitalZ”;
Both sizeof(str1) and sizeof(str2) will return the same value. (assume a 64-bit machine)

a) True
b) False

Q 9:

For the following two statements: char *str1 = “capitalZ”; and char str2[10] = “capitalZ”;
Both strlen(str1) and strlen(str2) will return the same value. (assume a 64-bit machine)

a) True
b) False

Q 10:

What is the primary difference between debugging and error handling?

a) Debugging involves runtime errors, while error handling is only compile-time.
b) Debugging finds errors, and error handling manages errors once identified.
c) Error handling prevents errors; debugging encourages them.
d) Debugging pauses errors; error handling triggers errors.

Q 11:

Which signal is raised when a program tries to write data to unallocated memory?

a) SIGTERM
b) SIGSEGV
c) SIGABRT

Q 12:

```
int func(void){
   static int x = 1;
   return ++x;
}
int main(){
   int y = func();
}
```

What will be the value of y after the last statement is executed?

a) 1
b) 2
c) 4
d) 5

In the following program (named reverse_string.c),
Four lines are missing in the program whose places are marked with the comments that start with //line 1, //line 2, …, //line 5. Answer the following 5 questions (from q33 to q37) about the program.

```
// reverse_string.c

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

char* read_input_and_reverse(int max_length) {
   // Line 1: Dynamically allocate memory for the input string
   char* input = /* missing allocation */;
   if (input == NULL) {
      perror("Memory allocation failed");
      return NULL;
   }

   printf("Enter a string (max length %d): ", max_length);
   // Line 2: Read input from the user into the allocated memory

   int length = strlen(input);

    char* reversed = (char*)malloc(strlen(input) + 1);
   if (reversed == NULL) {
      perror("Memory allocation failed for reversed string");
      free(input); // Free input memory if allocation fails
      return NULL;
   }

   for (int i = 0; i < length; i++) {
            // Line 3: Reverse the input string and store it in reversed

   }
   reversed[length] = '\0'; // Null-terminate the reversed string

   free(input); // Free the original input memory
   return reversed;
}

int main() {
   int max_length = 50;
   // Line 4: Call read_input_and_reverse and store the result in a pointer
   char* reversed_string = /* missing function call */;

   if (reversed_string == NULL) {
      return -1; // Exit if memory allocation fails
   }

   printf("Reversed string: %s\n", reversed_string);

   // Line 5: Free the memory allocated for the reversed string
   return 0;
}
```

Q 13:

What is the code line that should replace the comment // Line 1?
The purpose of this line is to dynamically allocate memory for the input string.

a) char* input = malloc(max_length * sizeof(char));
b) char* input = (char*)malloc(max_length);
c) char* input = calloc(max_length, sizeof(char));
d) All of the above

Q 14:

What is the code line that should replace the comment // Line 2?
The purpose of this line is to read input from the user into the allocated memory.
(Note: The input may have spaces)

a) scanf("%s", input);
b) sscanf (input, “%s”);
c) fgets(input, max_length, stdin);
d) None of the above

Q 15:

What is the code line that should replace the comment that starts with //line 3?
The purpose of this line is to reverse the input string and store it in reverse.

a) reversed[i] = input[length - i - 1];
b) reversed[length - i - 1] = input[i];
c) reversed[i] = input[i];

Q 16:

What is the code line that should replace the comment that starts with //line 4?
The purpose of this line is to call the read_input_and_reverse function and store the result in a pointer.

a) char* reversed_string = read_input_and_reverse(max_length);
b) char* reversed_string = read_input_and_reverse(strlen(input));
c) char* reversed_string = malloc(max_length);
d) None of the above

Q 17:

What is the code line that should replace the comment that starts with //line 5?
The purpose of this line is to free the memory allocated for the reversed string.

a) free(reversed_string);
b) free(input);
c) input = NULL;
d) reversed_string = NULL;

Q 18:

The line: char* input = malloc(100);
can be replaced with the line: char* input = realloc(NULL, 100);

a) True
b) False

Q 19:

What does the following command do to the files inside ../work?
find ../work -type f -empty -atime +30 | xarg rm

a) Removes empty directories
b) Removes empty regular files that have not been accessed more than 30 days
c) Removes empty directories that have not been accessed more than 30 days
d) The command will not work (has no effect on files inside ../work)

Q 20:

The following two lines represent two declarations of string variables:

- char* str1 = malloc(100);
- char str2[100];

Both variables can be used to read a string from the user.

a) True
b) False

Q 21:

The function call fseek(file, 0, SEEK_CUR); sets the file position at the same location as rewind(file). (file is the file handler)

a) True
b) False

Q 22:

If the file does not exist, what will happen when the fopen() function is called with the "r+" mode?

a) The file will be created.
b) fopen() will return a NULL pointer.
c) The program will wait for user input.
d) The program will throw an error.

Q 23:

What will the following statement do?

unsigned short int result = ((N | (1 << k))>>k) & 1;

a) Set the k-th bit and save its new value in the variable result.
b) Clear the k-th bit and save its new value in the variable result.
c) Toggle the k-th bit and save its new value in the variable result.