Patuakhali Science and Technology University

Course CIT 111

Submitted to:

Md. Mahbubur Rahman

Submitted by:

Md. Sharafat Karim

Id: 2102024

Reg: 10151

Assignment: 05

Title: Chapter 5 (theory)

Chapter 5

MULTIPLE CHOICE QUESTIONS

5.1 String constants are enclosed within ___ symbols

Ans: (b) double quote

5.2 Which of the following is used to ~~sp~~ separate specifiers in a format string within a scanf statement?

Ans: ~~(b) comma~~ (a) single white space

5.3 Which of the following functions can be used to print a single character as output?

Ans: ~~(c) putchar~~ (d) Both 2 and 3

5.4 Which of the following statements are true about scanf() function?

Ans: ~~(c) Any unread data items are considered for next call~~ (d) All of the above are true

5.5 Which of the following statements is false about print() function?

Ans: (d) All of the above are true

5.6 Which of the following format specifiers can be used to read floating type values?

Ans: (d) All of the above

REVIEW QUESTIONS

5.1 (a) The purpose of the header file <stdio.h> is to store the programs created by users. [False]

(b) The C standard function that recives a single character from the keyboard is getchar. [True]

(c) The input list in a scanf statement can contain one or more variables. [True]

(d) The scanf function cannot be used to read a single character from the keyboard. [False]

(e) The getchar cannot be used to read a line of text from the keyboard. [True]

(f) Variables from a legal element of the format control string of a printf statement. [True]

(g) The format specification %+ -8d prints an integer left-justified in a field width of 8 with a plus sign, if the number is positive. [True]

(h) If the field width of a format specifier is larger than the actual width of the value, the value is printed right-justified in the field. [True]

(i) The format specification %5s will print only the first 5 characters of a given string to be printed. ~~True~~ [False]

(j) When an input stream contains more data items than the number of specifications in a scanf statement, the unused items will be used by the next scanf call in the program. [True]

(k) Format specifiers for output convert internal representations for data to readable characters. [True]

(l) The print list in a printf statement can contain function calls. [True]

5.2 (a) The %hd specification is used to read or write a short integer.

(b) For reading a double type value, we must use the specification %lf

(c) For using characters functions, we must include the header file ctype.h in the program.

(d) To print the data left justified, we must use "-" in the field specification.

(e) The conversion specifier %x is used to print integers in hexadecimal form.

(f) The specification % is used to ~~print integers~~ read a data from input list and discard it without assigning it to many variables.

(g) The specification %[ ] is used for reading strings that contain characters.

(h) By default, the real numbers are printed with a precision of one decimal places.

(i) The specifier .%~~f~~e prints floating-point values in the scientific notation.

(j) The specification %[^] may be used in scanf to terminate reading at the encounter of a particular character.

(k) %c and %s format specifiers are used to print a single character and string respectively.

(l) Long integers are printed by specifying %ld in the place of d in the format specification. Likewise, short integers are printed using %hd.

5.3 (a) getchar only takes one character as an input. On the other hand scanf function can take any data type as an input.

(b) %s is used to read string or multiple characters. While %c is used to read just a single character.

(c) %f will print values with floating point without exponent. Where %g is more dynamic which can print both exponent and non-exponent depending on the value.

(d) %s is used for reading strings. On the other hand %[ ] is also used to read a string. But %s will be terminated if it finds a white-space but %[ ] can be customized under different conditions.

(e) %f is used to print floating point without exponent. And on the other hand %e is also used to print floating point values but it'll include exponent.

5.4 (a) 78 B 45

Ans: scanf ("%d %c %d", &a, &b, &c);

(b) 123 1.23 45A

Ans: scanf ("%d %f %s", &a, &b, c);

(c) 15-10-2002

Ans: scanf ("%s", a);

(d) 10 TRUE 20

Ans: scanf ("%d %s %d", &a, b, &c);

5.5 (a) printf ("%d %c %f", 10, 'x', 1.23);

Ans: 10 x 1.230000

(b) printf ("%2d %c %4.2f", 1234, 'x', 1.23);

Ans: 1234 x 1.23

(c) printf ("%d\t%4.2f", 1234, 456);

Ans: 1234 0.00

(d) printf ("\"%08.2f\"", 123.4);

Ans: "00123.40"

(e) printf ("%d %d%d", 10, 20);

Ans: 1020 -149958632

5.6 What will be values stored in the variable year and code when the data 1988, x is keyed in as a response to the following statements

(a) scanf ("%d %c", &year, &code);

Ans: year = 1988

code = x

(b) scanf ("%c %d", &year, &code);

Ans: year = -369098749

code = ? (invalid character)

(c) scanf("%d %c", &code, &year);

Ans: year = 2063597688

code = ◇ (invalid character)

(d) scanf("%s %c", &year, &code);

Ans: year = 243208753

code = x

5.7 The variables count, price and city have the following values:

count ← 1275

price ← -235.74

city ← Cambridge

(a) printf("%d %f", count, price);

Ans: 1275 -235.740005

(b) printf("%-2d\n%f", count, price);

Ans: 1275

-235.740005

(c) printf("%d %f", price, count);

Ans: 398976120 0.000000

(d) printf ("%10d xxxx %5.2f", count, price);

Ans: 1275 xxxx -235.74

(e) printf("%s", city);

Ans: Cambridge

(f) printf ("%-10d %-15s", count, city);

Ans: 1275 Cambridge

5.8 (a) printf (%d 7.2 %f, year, code);

Ans: Semicolon isn't used and 7.2 is used outside of format specifier.

(b) printf("%-s, %c"\n, city, code);

Ans: \n is placed outside semicolon, thus it may not work.

(c) printf ("%f, %d, %s, price, count, city);

Ans: Semicolon was not ended.

(d) printf("%c %d %f\n", amount, code, year);

Ans: ~~No~~ This is correct.

5.9 In response to the input statement

```
scanf ("%4d %*%d", &year, &code, &count);
```

the following data is keyed in

19 883715

What does the computer assign to the variables year, code and count?

Ans: year = 1988, count = 0

other variables will have garbage value.

5.10 How can we use the getchar() function to read multicharacter strings?

Ans: We can use a loop to use getchar again and again to read multicharacter strings. For example,

```
# include <string.h>
# include <stdio.h>
int main ()
{
    char c;
```

```
while ((c = getchar()) != '\n')
{
    printf("%c", c);
}
}
```

5.11 To use putchar() function to output multicharacter strings we can use a while loop until we find '\0'(Null).

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[ ] = "Hi there!";
    int i = 0;
    while (str[i] != '\0')
        putchar(str[i++])
}
```

5.12 Scanf is a standard input function to take input from a standard device to any memory address or a certain variable. It's defined inside stdio.h.

5.13 The commonly used conversation characters in a scanf function ~~are one to take a character~~ is to tell the computer what kind of data is given as an input.

5.14 (a) When more characters are given than the specific field width extra characters will go to the next input call.

(b) If input data has fewer characters than the data will work without any modification.

5.15 The purpose of printf function is to print something on a standard output device such as monitor. It is defined inside stdio.h header file.

5.16 The purpose of commonly used conversion characters in a printf function is to tell the compiler what kind of data is being inserted inside the string.

5.17 Control strings are both used in printf and scanf function. But in the printf function they are used to take a value and put it on the screen. On the other hand scanf use these control string to identify what kind of data is coming from a standard input device.

5.18 (a) If an output data item contains more characters than the specific field width then full of them will be printed.

(b) If an output data item contain fewer characters then blank white spaces are added in the output to makeup for the gap.

5.19 (a) In the scanf function, unrecognized characters within the control string are treated as delimiters. This means that when scanf encounters an unrecognized character in the control string it will stop reading input

(b) Unrecognized characters will be shown in the output. It may even contain garbage values.

DEBUGGING EXERCISES

(a) scanf ("%c %f %d", city, &price, &year);

Ans: Ampersand (&) is missing before city.

(b) scanf ("%s%d", city, amount);

Ans: Ampersand (&) is missing before amount

(c) scanf ("\n" %f", root);

Ans: '\n' should be placed inside the string.

(d) scanf ("%f %d", &amount, &year);

Ans: No error.

(e) scanf ("%c %d %ld", code, &count, Root);

Ans: Ampersand (&) is missing before Root.

INTERVIEW QUESTIONS

5.1 To print % character on the output string we have to use \%

5.2 The code will output 1 the size of char.

5.3 No, the following statement will not compile successfully. There is a right parenthesis missing.

5.4 The following code won't run successfully. getch and printf requires string.h and stdio.h.

5.5 The output will be any random garbage value.

5.6 The output will be 20.