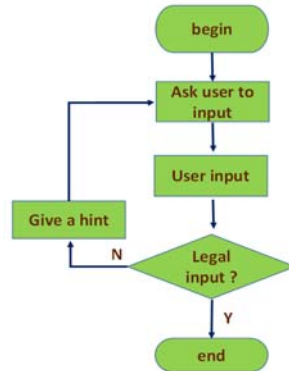


Practice 3-1 (2016/10/4)

Friendly and Smart Interactive Query – (II)

1. Another Instruction *getchar()* to Read a Character from stdin

A. Nightmare



```
alumni.cs.nctu.edu.tw - PuTTY
#include<stdio.h>
int main() {
    int val;
Lab1:  printf("Please input a number that is larger than 10 and less than 100: ");
    scanf("%d",&val);
    if (val < 10 || val > 100) {
        printf("Your input number %d is %s and it is an illegal input.\n",val, (val <=
10) ? "less than 10" : "larger than 100");
        goto Lab1;
    }
    printf("Your input number is %d, this is a valid value.\n", val);
    return 0;
}
```

```
alumni.cs.nctu.edu.tw - PuTTY
ylli@linux1: [~/prg]$ ./test
Please input a number that is larger than 10 and less than 100: 9
Your input number 9 is less than 10 and it is an illegal input.
Please input a number that is larger than 10 and less than 100: 122
Your input number 122 is larger than 100 and it is an illegal input.
Please input a number that is larger than 10 and less than 100: a
```

```
alumni.cs.nctu.edu.tw - PuTTY
r than 100 and it is an illegal input.
Please input a number that is larger than 10 and less than 100: Your input number 122 is large
r than 100 and it is an illegal input.
Please input a number that is larger than 10 and less than 100: Your input number 122 is large
r than 100 and it is an illegal input.
Please input a number that is larger than 10 and less than 100: Your input number 122 is large
r than 100 and it is an illegal input.
Please input a number that is larger than 10 and less than 100: Your input number 122 is large
r than 100 and it is an illegal input.
Please input a number that is larger than 10 and less than 100: Your input number 122 is large
r than 100 and it is an illegal input.
Please input a number that is larger than 10 and less than 100: Your input number 122 is large
r than 100 and it is an illegal input.
Please input a number that is larger than 10 and less than 100: Your input number 122 is large
r than 100 and it is an illegal input.
```

☞ The above program will repeat the message display and you have no chance to input data if you input some non-numerical character → characters in keyboard buffer are not removed by *scanf()*.

B. Scavenger for keyboard buffer, *getchar()*

☞ You should already find *scanf()* does not receive invalid input characters and leave these invalid characters in keyboard buffer. As you try to use

`scanf()` to start a new query, it easily gets into an infinite loop and you have no chance to input a character any more. Use `getchar()` to consume invalid characters of previous input. After that you can start a new query again.

```
alumni.cs.nctu.edu.tw - PuTTY
#include <stdio.h>

void main(void)
{
    int num, cnt;
L1: printf("Please input an integer that is larger than 10 and less than 100:");
    cnt = scanf("%d",&num);
    printf("We receive %d integer of %d from you.\n",cnt ,cnt ? num : 0);
    if (cnt != 1 || num <= 10 || num >= 100) {
        char ch = getchar();
        printf("ch = %c\n", ch);
        if (ch != '\n')
            printf("ch = %c\n",ch = getchar());
        goto L1;
    }
    printf("Input complete!\n");
}
```

```
alumni.cs.nctu.edu.tw - PuTTY
ylli@linux1:[~/prg]$ readInt
Please input an integer that is larger than 10 and less than 100:200
We receive 1 integer of 200 from you.
ch =
Please input an integer that is larger than 10 and less than 100:200a
We receive 1 integer of 200 from you.
ch = a
ch =
Please input an integer that is larger than 10 and less than 100:200ab
We receive 1 integer of 200 from you.
ch = a
ch = b
Please input an integer that is larger than 10 and less than 100:200abc
We receive 1 integer of 200 from you.
ch = a
ch = b
Please input an integer that is larger than 10 and less than 100:We receive 0 integer of 0 from you.
ch = c
ch =
Please input an integer that is larger than 10 and less than 100:200a
We receive 1 integer of 200 from you.
ch = a
ch =
Please input an integer that is larger than 10 and less than 100:
```

Hint: You can use `getchar()` to remove every character in keyboard buffer until `getchar()` returns a newline character.

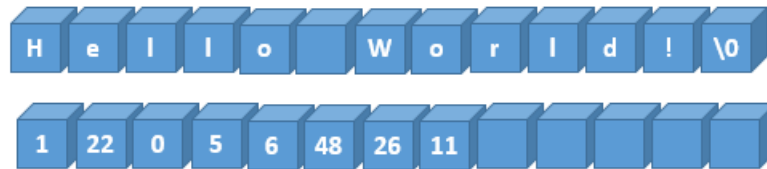
2. String

A. Array Definition

☞ A group of consecutive data of the same type, for instance, integer array and character array (string).

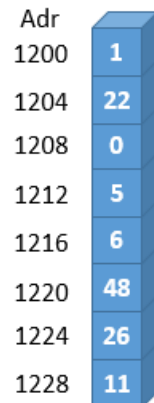
☞ store cell phone number of my friends (5 friends) in an array.

```
int phNo1, phNo2, phNo3, phNo4, phNo5;
phNo1 = 0918293843; phNo2 = 0937193227;
int phNo[5]; // phNo[0] to phNo[4]
phNo[0] = 0918293843; phNo[1] = 0937193227;
```



B. Array Location

☞ In memory, an array is stored in consecutive locations.



C. String

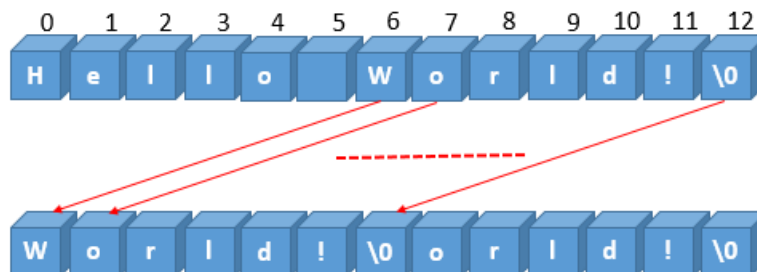
☞ A string is a sequence of characters ending with null character '\0'.

☞ We use null character to detect where a string ends.

D. String copy & cutting

☞ Ex. `char str[50], cpStr[50];`

`cpStr[0] = str[0]; cpStr[1] = str[1]; ...; cpStr[12] = str[12];`



☞ `str[0] = str[6]; str [1] = str[7]; ...; str[6] = str[12];` (self-cutting) or

`cpStr[0] = str[6]; cpStr[1] = str[7]; ...; cpStr[6] = str[12];` (copy and cut)

☞ **Better way to perform iterative operations:** for `i = 0` to `6` do `str[i] = str[i+6];`

In C **for** loop can complete this kind of iterative operations.

for (`i = 0`; `i <= 6`; `i = i + 1`)

`str[i] = str[i+6]; /* or cpStr[i] = str[i+6]; */`

----- **or** -----

`i = 0;`

while (`i <= 6`) {

`str[i] = str[i+6]; /* or cpStr[i] = str[i+6]; */`

```

        i = i + 1;
    }

```

```

alumni.cs.nctu.edu.tw - PuTTY
#include <stdio.h>
#include <string.h>

int main(void)
{
    char str1[50], str2[50] = "Hello NCTU CS!";
    printf("str1 = \"%s\\\", and str2 = \"%s\\\"\\n\",str1, str2);

    str1[0] = 'B'; str1[1] = 'i'; str1[2] = 'g'; str1[3] = ' ';
    str1[4] = 'b'; str1[5] = 'a'; str1[6] = 'n'; str1[7] = 'g'; str1[8] = '\\0';
    printf("str1 = \"%s\\\", and str2 = \"%s\\\"\\n\",str1, str2);
    strcpy(str1, str2);
    printf("str1 = \"%s\\\", and str2 = \"%s\\\"\\n\",str1, str2);
    for (int i = 0; str1[i] != '\\0'; i++) str1[i] = str1[i+1];
    printf("str1 = \"%s\\\", and str2 = \"%s\\\"\\n\",str1, str2);

    return 0;
}

```

```

alumni.cs.nctu.edu.tw - PuTTY
ylli@linux1: [~/prg]$ str
str1 = "", and str2 = "Hello NCTU CS!"
str1 = "Big bang", and str2 = "Hello NCTU CS!"
str1 = "Hello NCTU CS!", and str2 = "Hello NCTU CS!"
str1 = "ello NCTU CS!", and str2 = "Hello NCTU CS!"
ylli@linux1: [~/prg]$

```

3. Problem 1

Last time we already realized how to read data in different type with *scanf()*, but failed to read two integers with *scanf()* if users input non-numerical characters or string intentionally. Write a program to read two integers with a well-designed interactive query that prompts useful message to lead users to the right way of offering correct data. Remember to clear the screen before starting a new iteration of query (right after pressing any key to continue).

- a. Begin a query for two integers.

```

Please input two numbers:22 101

```

- b. Prompt two messages for a correct input.

```

-bash
Please input two numbers:22 101
The legal number of inputs are 2
Your input is completely correct!
Press any key to continue!

```

- c. After users press a key, the program returns to initiate a new query.

```

Please input two numbers:ss 11 d33 d_

```

- d. The input program cannot recognize any integer and prompts three messages

for users.

```
-bash
Please input two numbers:ss 11 d33 d
The legal number of inputs are 0
We do not receive two integers.
The garbage in keyboard buffer is "ss 11 d33 d". We just removed them from the buffer.
Press any key to continue!
```

e. The program returns to initiate a new query after users press any key.

```
Please input two numbers:333 sd2 e 32
```

f. The program recognizes one integer and prompts three messages.

```
-bash
Please input two numbers:333 sd2 e 32
The legal number of inputs are 1
We do not receive two integers.
The garbage in keyboard buffer is "sd2 e 32". We just removed them from the buffer.
Press any key to continue!
```

g. The program initiates a new query after users press any key.

```
Please input two numbers:987 348djed du3ed 343
```

h. The program recognizes two integers but there are remaining garbage messages.

```
-bash
Please input two numbers:987 348djed du3ed 343
The legal number of inputs are 2
Although we got two integers. You also input some non-numerical characters.
The garbage in keyboard buffer is "djed du3ed 343". We just removed them from the buffer.
Press any key to continue!
```

Hint: You can use `system("clear")` to clear screen in a C program. Before using `system`, you have to include `stdlib.h` by directive.

☞ Use `scanf()` and `getchar()` only without any flush statement to realize a friendly and smart interactive query program to accept two integers from users.

☞ Your program must work as steps a, b, c and d before the end of this practice time.

☞ The deadline to upload the program is the midnight of 10/10.

