

Announcements

Thursday Extras: CS Commons on Thursdays @ 4:00 pm
this week: ????

Office hours: normal this week

Monday: 1:00-3:00 pm

Tuesday, Thursday: 2:30-4:00 pm

Wednesday, Friday: 1:30-3:00 pm

Reminder: Supplemental Problem 2 due today, but
no penalty if submitted by 5:00 pm on Tuesday

Reflections on black-box and white-box testing

strings and scanf

questions

clicker questions

Clarifications about White-box and Black-box Testing

Consider these statements:

1. Black-box Testing only involves looking at the output given for specific inputs.
2. White-box Testing only involves looking at the output given for specific inputs.

Which, if any, of these statements is true?

- | | |
|-----------------|--------------------|
| A. Both 1 and 2 | C. 2, not 1 |
| B. 1, not 2 | D. neither 1 nor 2 |

Clarifications about White-box and Black-box Testing

Consider the following program outline:

Enter values

If (condition 1)

A

else

B

if (condition 2)

C

else

D

if (condition 3)

E

else

F

Assume conditions 1, 2 and 3
are independent of each other

Vote 1: What is the minimum
number of test cases required
for White-box testing?

A. 0-3

B. 4-6

C. 7-9

D. ≥ 10

E. cannot tell

Vote 2: What is the minimum
for Black-box testing?

A. < 8

B. $= 8$

C. > 8

D. no way to know

Consider the following program that reads 3 characters:

```
#include <stdio.h>
int main ()
{
    char a, b, c;
    scanf ("%c%c%c", &a, &b, &c);
    printf ("#%c#%c#%c#\n", a, b, c);
    return 0;
}
```

A user enters x y z (that is, the letters x, y, z, each separated by a space character; there is no space before the x or after z)

What is printed?

- A. compile error
- B. runtime error
- C. #x#y#z#
- D. #x# #z#
- E. something else

Consider the following program that reads 3 characters:

```
#include <stdio.h>
int main ()
{
    char a, b, c;
    scanf ("%c %c %c", &a, &b, &c); // spaces added before %c
    printf ("#%c#%c#%c#\n", a, b, c);
    return 0;
}
```

A user enters x y z (that is, the letters x, y, z, each separated by a space character; there is no space before the x or after z)

What is printed?

- A. compile error
- B. runtime error
- C. #x#y#z#
- D. #x# #z#
- E. something else

A code segment is to be written that will read a word (up to 10 characters) and print the result. The basic code progression is

Declaration of str variable

```
scanf ("%11s", str);  
printf ("the string is %s\n", str);
```

Five declarations are proposed:

1. char str[15];
2. char * str;
3. char word[15];
char * str = word
4. char word[15];
char str [] = word;
5. char str [15] = "my string"
6. char * str = "my string"

The user wants to type the word Hello.

When the code is compiled and run, either an error occurs or it runs fine:

6 Votes with these options:

- A. compilation error
- B. run-time error
- C. runs fine

6 Votes:

Vote 1: declaration 1

Vote 2: declaration 2

Vote 3: 2 lines from code 3

Vote 4: 2 lines from code 4

Vote 5: declaration 5

Vote 6: declaration 6

A user is to type words on each of 2 lines.

The program is to read the first word of each line, and ignore the rest of the input.

Assume each word is no longer than 9 characters, and assume the following declarations:

```
char str1 [10], str2 [10], bigstr[100], ch;
```

Two approaches are proposed:

1. `scanf ("%9s", str1);`
 `fgets (bigstr, 100, stdin);`
 `scanf ("%9s", str2);`
2. `scanf ("%9s", str1);`
 `while ((ch=getchar()) != '\n') ;`
 `scanf ("%9s", str2);`

Which, if any, of these approaches works?

A. Both 1 and 2

B. 1, not 2

C. 2, not 1

D. neither 1 nor 2

As in the previous slide, a user is to type a word on each of 2 lines. Two more approaches are proposed:

```
3.  int i = 0;
    do { str1[i] = getchar();
        i++;
    }
    while (!isspace(str1[i]))
    str1[i] = 0;
    while ((ch=getchar()) != '\n') ;
    int i = 0;
    do { str2[i] = getchar();
        i++;
    }
    while (!isspace(str2[i]))
    str2[i] = 0;
```

```
4.  int i = 0;
    do { str1[i] = getchar();
        i++;
    }
    while (!isspace(str1[i-1]))
    while ((ch=getchar()) != '\n') ;
    int i = 0;
    do { str2[i] = getchar();
        i++;
    }
    while (!isspace(str2[i-1]))
```

- A. Both 3 and 4
- B. 3, not 4

- C. 4, not 3
- D. neither 3 nor 4

Suppose input will contain a name in the first 40 characters of a line, followed by a number.

```
char line [101]; //leaving room for null character
char name [41];
int number;
```

Two approaches are proposed reading a line

1. `scanf ("%41s %d", name, &number);`
2. `fgets (line, 101, stdin);`
 `strncpy (name, line, 40);`
 `name[40] = 0;`
 `number = atoi (line+40);`

Which of these approaches works reliably?

- A. Approaches 1 and 2
- B. Approach 1, not 2
- C. Approach 2, not 1
- D. Neither 1 nor 2

Suppose you want to read an integer at the start of one line, skip anything else on that line, and then read a double at the start of the next line.

```
int one;  
double two;  
#define max 100  
char line [max+1];
```

Two approaches are proposed:

1. `scanf ("%d", &one);`
 `while (getchar () != '\n') ;`
 `scanf ("%lf", &two);`
2. `fgets (line, max+1, stdin);`
 `one = atoi (line);`
 `while (line[strlen(line)-1] != '\n')`
 `fgets (line, max+1, stdin);`
 `fgets (line, max+1, stdin);`
 `two = atof (line);`

Which of these approaches works reliably?

- A. Approaches 1 and 2
- B. Approach 1, not 2
- C. Approach 2, not 1
- D. Neither 1 nor 2