

ESC101N: Fundamentals of Computing  
Mid-sem 1  
2010-11 1st semester

Instructor: Arnab Bhattacharya

2:00-3:00pm, 31st August, 2010

## Instructions

1. Please write your name, roll number and section below.
2. Please write your roll number and section in each page of the question paper.
3. Please hand over the paper after the exam.
4. This question paper contains 5 questions in 6 pages.

**Name:**

**Roll Number:**

**Section:**

Question	1	2	3	4	5	Total
Marks	10	10	10	10	10	50
Score						

1. State whether the following expressions are valid. Assume that the values of the variables  $i$  and  $j$  are 3 and 2 respectively before the evaluation of each expression. If it is valid, evaluate the value. If it is not valid, state the reason.

- [2] (a)  $i++*--j$   
 [1] (b)  $3+-5*-2$   
 [1] (c)  $3>5-2$   
 [1] (d)  $3*5/2$   
 [2] (e)  $++3*5$   
 [1] (f)  $(i>j)\&\&(i=j)$   
 [2] (g)  $i+3\%j-1$

**Answer:**

- (a) valid, 3  
 (b) valid, 13  
 (c) valid, 0  
 (d) valid, 7  
 (e) invalid, ++ operator works on a variable, not value  
 (f) valid, 1  
 (g) valid, 3

- [10] 2. Given an array of 5 real numbers, the following program finds the harmonic mean of them. For segments of the program marked as `??1??` to `??7??`, appropriately fill in the code. The harmonic mean HM of  $n$  numbers  $x_1, \dots, x_n$  is given by the equation

$$\frac{1}{HM} = \frac{1}{n} \left( \frac{1}{x_1} + \dots + \frac{1}{x_n} \right)$$

```
.
#include <stdio.h>
int main()
{
    double a[5];
    double hm, sum;
    int i;

    sum = ??1??;
    for (??2??; ??3??; ??4??)
    {
        scanf("??5??", &a[i]);
        sum = sum + ??6??;
    }
    hm = ??7??;
    printf("HM = %lf\n", hm);
}
```

**Answer:**

??1??: 0 [1]

??2??: i = 0 [1]

??3??: i &lt; 5 [2]

??4??: i++ [1]

??5??: %lf or %f [1]

??6??: (1 / a[i]) [2]

??7??: 5 / sum [2]

3. (a) Study the following program carefully. Assume that  $-102$  has been input. The power function raises the first integer to the power of the second integer and returns the value. For example, `power(3, 2)` evaluates to  $3^2 = 9$ .

- [2] i. What is the output of the program?
- [1] ii. What is the value of `i` after line 14 has been executed?
- [1] iii. What is the value of `d` after line 22 but before line 23 has been executed?
- [1] iv. What does `d` signify?
- [2] v. What is the value of `j` after line 31 has been executed in each iteration of the loop?

```
1 #include <stdio.h>
2 int main()
3 {
4     int i, j, k, d;
5     char c;
6
7     scanf("%d", &i);
8     i = i + 5;
9
10    if (i < 0)
11    {
12        putchar('-');
13        i = -i;
14    }
15
16    j = i;
17    d = 0;
18    while (j > 0)
19    {
20        d++;
21        j = j / 10;
22    }
23    d--;
24
25    j = i;
26    while (j > 0)
27    {
```

```

28         k = j / power(10, d);
29         c = k + '0';
30         putchar(c);
31         j = j - k * power(10, d);
32         d--;
33     }
34
35     if (i == 0)
36         putchar('0');
37 }

```

- [3] (b) For any general integer  $i$ , what does the program do?

**Answer:**

(a) i. -97

ii. 97

iii. 2

iv.  $d$  stores the number of digits.

v. In first iteration, value of  $j$  is 7.

In second iteration, value of  $j$  is 0.

(b) The program inputs an integer from the user, adds 5 to it, and then prints the significant digits of the resulting integer as a sequence of characters (including the '-' symbol).

- [10] 4. The following program checks whether a password is valid. The password is always 8 character long. It is stored as an array of characters in `password`. A password is valid if it contains at least one digit, one letter and one special character. The set of special characters is '#' and '@'. Fill in the blanks `??1??` to `??2??` appropriately to complete the program.

```

#include <stdio.h>
int main()
{
    char password[8];
    int i;

    int digit = 0;           // indicates if there is a digit
    int letter = 0;         // indicates if there is a letter
    int special = 0;        // indicates if there is a special character

    for (??1??; i < 8; i++)
        password[i] = getchar();

    for (i = 0; ??2??; i++)
    {
        if ((password[i] >= '0') ??3?? (password[i] <= '9'))
            digit = 1;
    }
}

```

```

        if ((password[i] >= 'a') && (password[i] <= 'z'))
            letter = ??4??;
        if (??5??)
            letter = 1;
        if (??6??)
            special = 1;
    }

    if (??7??)
        printf("Password is valid\n");
    else
        printf("Password is invalid\n");
}

```

**Answer:**

```

??1??: i = 0 [1]
??2??: i < 8 [1]
??3??: && [1]
??4??: 1 [1]
??5??: (password[i] >= 'A') && (password[i] <= 'Z') [2]
??6??: (password[i] == '#' || password[i] == '@') [2]
??7??: (digit == 1) && (letter == 1) && (special == 1) [2]

```

- [10] 5. The following program prints all natural numbers (i.e., positive integers) in the first array  $A$  that are absent in the second array  $B$  (similar to set difference  $A - B$ ). For example, if  $A = \{3, 4, 4, 7, 2, 8, 2, 5, 4, 1\}$  and  $B = \{4, 7, 9\}$ , then the program should print 3 2 8 2 5 1. Complete the code segment below by appropriately filling in the blanks ??1?? to ??7??.

```

#include <stdio.h>
int main()
{
    int A[10];
    int B[3];
    ??1?? i, j;

    for (i = 0; i < 10; i++)
        scanf("%d", &A[i]);
    for (j = 0; j < 3; j++)
        scanf("??2??", &B[j]);

    for (??3??; j < 3; j++)
        for (??4??; ??5??; i++)
            if (??6??)
                A[i] = 0;

    for (i = 0; i < 10; i++)

```

```
        if (????)
            printf("%d ", A[i]);
    printf("\n");
}
```

**Answer:**

??1??: int or char [1]

??2??: %d [1]

??3??: j = 0 [1]

??4??: i = 0 [1]

??5??: i &lt; 10 [2]

??6??: B[j] == A[i] [2]

??7??: A[i] != 0 [2]