

ESC101: Fundamentals of Computing
Mid Semester Exam (September 17, 2013)

Name:

Roll Number:

Section:

**I PLEDGE MY HONOUR AS A GENTLEMAN/LADY THAT DURING THE EXAMINATION I
HAVE NEITHER GIVEN ASSISTANCE NOR RECEIVED ASSISTANCE.**

.....
Signature

Duration: 2 hours

Total Marks: 200

Instructions

1. Marks will be deducted if you forget to enter your name, roll number, section or signature.
2. There are **NINE** questions in this question booklet on **TWELVE** pages. Do check that your booklet has all the problems.
3. You have to write in the space provided. Avoid overwriting.

Question	Marks
Q1	
Q2	
Q3	
Q4	
Q5	
Q6	
Q7	
Q8	
Q9	
Total	

1. State whether the following expressions are valid. For each case, if the expression is valid, evaluate the value; if the expression is not valid, state the reason. [Total: 24 marks]

Assume that the values of the variables `i` and `j` are 3 and 2 respectively before the evaluation of each expression. Also, assume that both `i` and `j` are of `int` type.

(a) `(3>5-2)`

Answer: Valid

Value of expression: 0

(b) `((i>j)&&(i=j))`

Answer: Valid

Value of expression: 1

(c) `(3*-5/2)`

Answer: Valid

Value of expression: -7

(d) `(++3*5)`

Answer: Invalid

Reason: Increment operator requires a variable to perform its operation, it cannot operate on numerical constants

(e) `(i = 2, -j)`

Answer: Valid

Value of expression: -2

(f) `(i<=3&& i==2)`

Answer: Valid

Value of expression: 0

(g) `(i, 31, 41)`

Answer: Valid

Value of expression: 41

(h) `(!i || j)`

Answer: Valid

Value of expression: 1

Grading Policy:

- 3 marks for each part
- 1 marks if the person has written whether the expression is valid or not
- Complete marks if for valid statements, the correct value is written

2. After each of the following code fragments, print the value of variable x. If there is any compilation error, mention the cause of the error. [Total: 20 marks]

(a) `float f = 354;`
`int x = 5/6 + 6/7 + (int)(f/10) + 9/10;`

Answer: 35

(b) `int b1 = 3;`
`int b2 = 4;`
`long long x = b1*b2;`

Answer: 12

(c) `float f = 3.1;`
`float x = (int)((f+3)/3*7/2);`

Answer: 7.000000

(d) `long t = 124;`
`float x = (int)((float)t/3 + 0.8);`

Answer: 42.000000

(e) `double d = 6.6;`
`long x = (long)(d*2+9/10);`

Answer: 13

Grading Policy

- All Parts carry equal marks (4)
- Partial Grading:
 - (a) If in parts b), c), d) print format of the data types are not correct (Int to Float/Float to Int), two marks are deducted. For part e), outputs other than 13 fetch zero marks.
 - (b) If in any of the parts, the student mentions words like "error"/ "invalid", no marks would be awarded.

3. Print the output of the following program when input given is:

[Total: 16 marks]

(a) 12

(b) 15

```
#include<stdio.h>

int main()
{
    int num;
    int count;
    count = 0;
    scanf("%d", &num);
    while(num>1)
    {
        if(num%2==0)
        {
            num = num/2;
        }
        else
        {
            num = 3*num + 1;
        }
        count++;
        printf("%d: %d\n",count, num);
    }
}
```

Answer (a):

1: 6
2: 3
3: 10
4: 5
5: 16
6: 8
7: 4
8: 2
9: 1

Answer (b):

1: 46
2: 23
3: 70
4: 35
5: 106
6: 53
7: 160
8: 80
9: 40
10: 20
11: 10
12: 5
13: 16
14: 8
15: 4
16: 2
17: 1

Grading Policy:

- Part (a) - 6 marks
- Part (b) - 10 marks

4. Consider the following program:

[Total: 20 marks]

```
#include <stdio.h>

int main()
{
    char c;

    //Enter the string that will be read by getchar() character by character till
    //it encounters '\n'

    while((c=getchar())!='\n')
    {
        if(((c>'A') && (c<='Z')) || ((c>'a') && (c<='z')))
            printf("%c", c-1);

        else if((c=='A') || (c=='a'))
            printf("%c", c+25);

        else
            printf("%c", c);
    }
    printf("\n");
}
```

Write down the the output of the code when the input is (each of the inputs is followed by a newline character):

- (a) ahHjD
- (b) Esc101MidsemExam
- (c) Esc101!_is_2EASY
- (d) \$Exam!IsNot!Hard\$

Answers:

- (a) zggGiC
- (b) Drb101LhcrdlDwzl
- (c) Drb101!_hr_2DZRX
- (d) \$Dwzl!HrMns!Gzqc\$

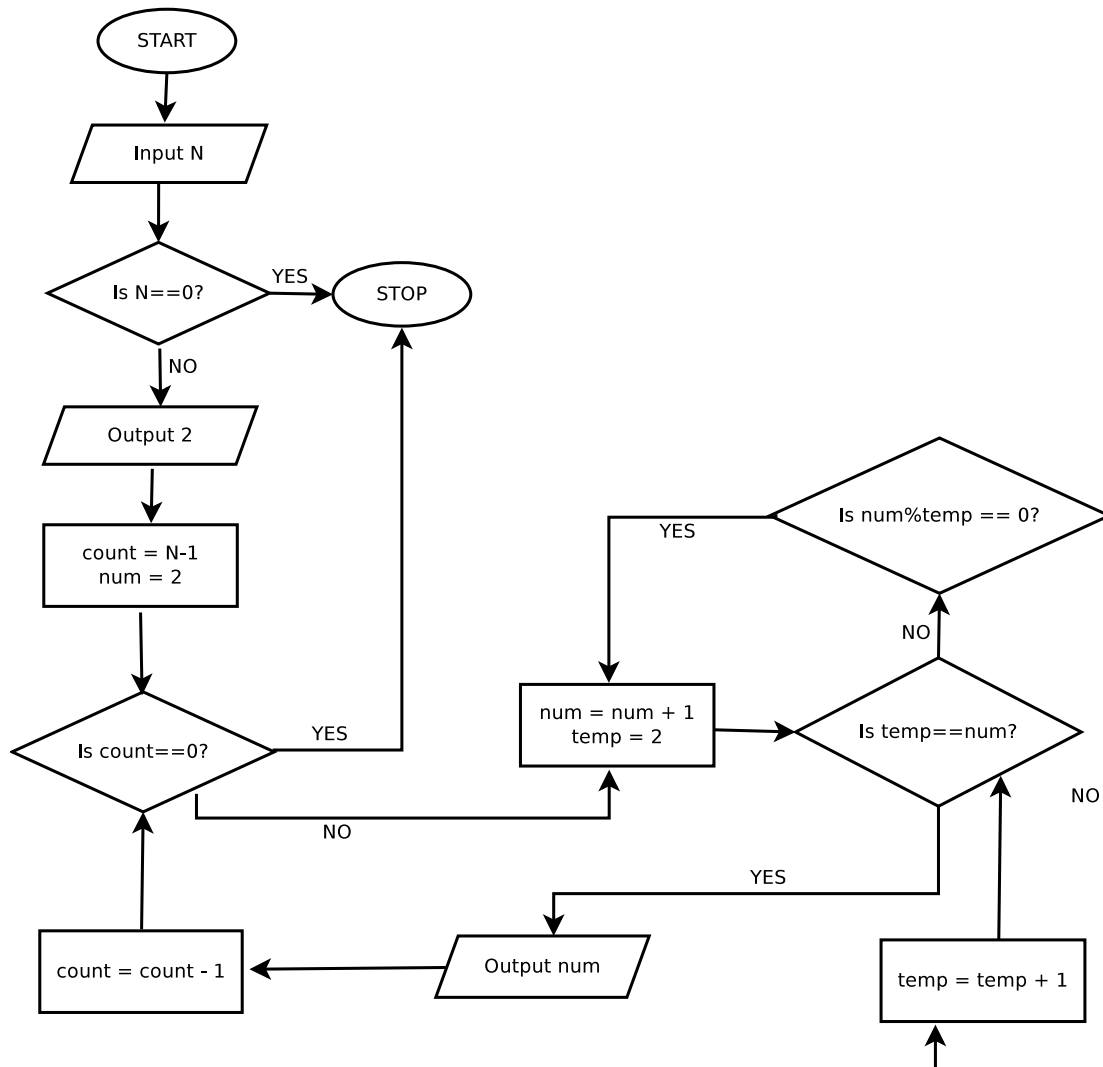
Grading Policy:

- Marks distribution - 18 (for correct output string) + 2 (for correct format, i.e. string written with all characters in a single line)
- Part (a) - 3 Marks + 0.5 Marks
- Part (b) to (d) - 5 Marks + 0.5 Marks each

5. Study the flowchart below and answer the following questions:

[Total: 30 marks]

- Print the output when $N = 6$.
- For how many different values of N (for $N \geq 0$) does the number 17259 appear in the output?
- When $N = 178$, the last number printed is 1061. What is the GCD of 178 and 1061?
- When N is given as the input, it also appears in the output; which of these can N possibly be (only one answer is correct): 3972, 3975, 459, 451, 17, 16, 15.



Answer (a):

2

3

5

7

11

13

Answer (c):

1

Answer (b):

0

Answer (d):

17

Grading Policy:

- Part a - 2(1 Marks), 3(3 Marks), 5(3 Marks), 7(3 Marks), 11(3 Marks), 13(3 Marks) = 16 Marks
- Part b - 5 Marks
- Part c - 4 Marks
- Part d - 5 Marks

6. Consider the following program:

[Total: 25 marks]

```
#include<stdio.h>

int main()
{
    int low=0,high=0;
    //Enter the range low and high
    scanf("%d %d", &low, &high);

    int i, digit, n, sum;

    for(i = low; i<= high ;i++)
    {
        n = i;
        sum = n;

        while(sum > 9)
        {
            n = sum;
            sum = 0;
            while(n > 0)
            {
                digit = n%10;
                n = n/10;
                sum = sum + digit;
            }
        }
        if(sum == 1)
        {
            printf(" %d is a ESC101MidSem1 number \n",i);
        }
    }
    return 0;
}
```

Write down the the output of the code when the input is:

- (a) 10 30
- (b) 50 100

Answers:

- (a) 10 is a ESC101MidSem1 number
19 is a ESC101MidSem1 number
28 is a ESC101MidSem1 number

- (b) 55 is a ESC101MidSem1 number
64 is a ESC101MidSem1 number
73 is a ESC101MidSem1 number
82 is a ESC101MidSem1 number
91 is a ESC101MidSem1 number
100 is a ESC101MidSem1 number

Grading Policy:

- Part a - 6 Marks
- Part b - 18 Marks
- Bonus - 1 Mark for all correct
- Partial marking: 2 marks for each correct line in part (a) and 3 marks for each correct line in part (b).
If the ordering is incorrect or there are too many errors, there is a penalty (left to the discretion of the examiner as there are too many cases to list).

7. Study the following program carefully. The function `power()` returns the result of raising the first integer to the power of the second integer. For example, `power(3, 2)` evaluates to $3^2 = 9$.

Assuming that -102 was entered as the input, answer the following questions:

[Total: 25 marks]

- (a) What is the output of the program?
- (b) What is the value of the variable `i` after line 14 has been executed?
- (c) What is the value of the variable `d` after line 22 but before line 23 has been executed?
- (d) What does the variable `d` signify (what is its purpose in the program)?
- (e) What are the value of the variable `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        printf("-");
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        printf("%c", c);
31        j = j - k * power (10 , d);
32        d --;
33    }
34
35    if (i == 0)
36        printf("0");
37    return 0;
38 }
```

Answers:

(a)

-97

(b)

97

(c)

2

(d)

Consider the value of variable i just after the execution of line 14. Variable d stores the number of digits in that value. d is later used to isolate the digits of that number.

(e)

7 in first iteration and 0 in second iteration.

Grading Policy:

- (a) total marks = 5, full marks for the correct answer "-97" and partial credits otherwise.
- (b) total marks = 5, full marks for the correct answer "97" and 0 for incorrect.
- (c) Total marks = 5, full marks for correct answer 2 and 0 for incorrect.
- (d) total marks = 4, full marks for correct answer "for the variable i after line 14, d stores the number of digits in i. d is used to isolate the digits of i later." 3 marks if the answer is "d stores the number of digits of i for the variable after line 14". 2 marks if the answer is "d stores the number of digits of i". Zero marks for others.
- (e) Total marks = 6. 3 marks for correct answer "7 during the first iteration" of the first part and 3 marks for the correct answer "0 during the second iteration" for the second part.

8. The following program checks whether a password is valid. The password is always 8 character long. A password is valid if it contains at least one digit, one letter and one special character (the set of special characters consists of '#' and '@').

Fill in the blanks [Blank 1] to [Blank 6] appropriately to complete the program. *Make sure that your answers are consistent with the comments provided in the program.* [Total: 20 marks]

```
# include <stdio.h>
int main ()
{
    char password;
    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([Blank 1]; i < 8; i++)
    {
        password = getchar ();

        /* Test if the character read is a digit */
        if ([Blank 2])
            digit = 1;

        /* Test if the character read is a small case letter i.e. a-z */
        if ([Blank 3])
            letter = 1;

        /* Test if the character read is a capital case letter i.e. A-Z */
        if ([Blank 4])
            letter = 1;

        /* Test if the character read is a special character i.e. # or @ */
        if ([Blank 5])
            special = 1;
    }

    if ([Blank 6])
        printf ("Password is valid\n");
    else
        printf ("Password is invalid\n");
    return 0;
}
```

Answers:

Blank 1

`i = 0`

Blank 2

`password <= '9' && password >= '0'`

Blank 3

`password <= 'z' && password >= 'a'`

Blank 4

`password <= 'Z' && password >= 'A'`

Blank 5

`password == '#' || password == '@'`

Blank 6

`digit==1 && letter==1 && special==1`

Grading Policy:

- Blank1 - 2 Marks
- Blank2 to Blank5 - 3 Marks each
- Blank6 - 6 Marks

9. Consider the following program:

[Total: 20 marks]

```
#include <stdio.h>

int val(int, int);

int main()
{
    int n, k;
    int ans;
    scanf("%d %d", &n, &k);

    ans = val(n, k);

    printf("%d\n", ans);

    return 0;
}

int val(int n, int k)
{
    int t, temp;
    int ans;
    ans = 0;
    while(n>=k)
    {
        temp = n/k;
        t = n%k;
        ans = ans + temp;
        n = temp + t;
        printf("%d %d\n", ans, n);
    }
    return ans;
}
```

Write down the the output of the code when the input is:

- (a) 5 3
- (b) 10 2

Answer (a):

1 3
2 1
2

Answer (b)

5 5
7 3
8 2
9 1
9

Grading Policy:

- a) $2.5+2.5+2=7$
- b) $2.5+2.5+2.5+2.5+2=12$
- +1 bonus for all correct including spaces, new lines and order (2.5 for both values correct, 1 for single correct)