

ESC 101: Fundamentals of Computing				Mid-sem Exam (22 Sep 2018)		
Name						75 marks Page 1 of 8
Roll No		Dept.		Section		

### Instructions:

1. This question paper contains 4 pages (8 sides of paper). Please verify.
2. Write your name, roll number, department and section **on every sheet** of this booklet.
3. Write your final answers neatly **with a blue/black pen**. Pencil marks may get smudged.

**Q1.** Write **T** or **F** for True/False (write **only in the box on the right hand side**) (18x1=18marks)

For this question,  $i = j = k = m = s = t = 0$  are **int** variables,  $p = q = r = 0.0$  are **double** variables, and  $a = b = c = d = '0'$  are **char** variables

1	The for loop <code>for(i = 0; 0; i++) j++;</code> will run for infinitely many iterations.	
2	The two statements <code>j = k++;</code> and <code>j = (m += 1);</code> will result in the exact same value being assigned to the variable <code>j</code>	
3	If we create and initialize a character array as <code>char e[] = "ESC101";</code> then the array will contain 6 elements.	
4	The two expressions <code>('0' == ('1' - 1))</code> and <code>(12 == (6 + '6'))</code> will generate the same value.	
5	The statement <code>for(p = 0.0; q &lt; 3; p++) q++;</code> will cause a compilation error.	
6	The following statement will print the word "Wow" (without the quotes) <code>switch(a){ case 0: printf("Nah"); case '0': printf("Wow"); }</code>	
7	Let <code>int n[] = {4,3,2,1,0};</code> Then the statements <code>printf("%d", n[n[2]++]);</code> and <code>printf("%d", n[++s]);</code> will print the same value.	
8	The statement <code>d = b % c;</code> will result in a compilation error.	
9	The following statement will result in the word "Bye" getting printed (without the quotes) <code>if(0.0) printf("Hi"); else printf("Bye");</code>	
10	It is not possible to store the exact value of $\pi = 3.1415926535897932384626433\dots$ in a double variable such as <code>r</code> .	
11	The following statement will print the word "Wow" getting printed (without quotes) <code>switch(t){ case 0: printf("Nah"); case 0: printf("Wow"); }</code>	
12	Enumerations cannot be used to assign human-readable names to floating point constants such as 2.3 or 3.141	
13	Let <code>x, y</code> be two integer variables. Then the expression <code>(x = y)</code> will always be evaluated by Mr. C as true.	
14	The function <code>getchar()</code> will never read the newline character <code>'\n'</code> from the input.	
15	The statement <code>putchar('a'+0);</code> will print the character "a" (without quotes) to the output.	
16	The statement <code>printf("%d", 25 % ('e' - 'a'));</code> will print "0" (without quotes) to the output	
17	Flag variables can very well be defined as <b>float</b> variables if it suits the program.	
18	Negative floating numbers e.g <code>-2.5</code> are considered by Mr C to stand for <b>FALSE</b>	

**Q2.** Fill the circle (**don't tick**) next to the correct option (**only one choice correct**). (6x2=12marks)

**2.1** Which values of **a**, **b**, **c** will result in a true evaluation for the statement (**a < b < c**)

<b>A</b>	<b>a = -1, b = 0, c = 1</b>	<input type="radio"/>
<b>B</b>	<b>a = -30, b = -20, c = -10</b>	<input type="radio"/>
<b>C</b>	<b>a = 30, b = 20, c = 10</b>	<input type="radio"/>
<b>D</b>	<b>Both A and B</b>	<input type="radio"/>

**2.2** Which of the following statements is correct?

<b>A</b>	ASCII characters are numbered from 0 to 128	<input type="radio"/>
<b>B</b>	The statement <code>int x = 'z';</code> will cause a compilation error	<input type="radio"/>
<b>C</b>	The statement <code>char x = 97;</code> will cause a compilation error	<input type="radio"/>
<b>D</b>	None of the above	<input type="radio"/>

**2.3** If we declare `enum{pp = 5, qq, rr, ss};` then which of the following is true?

<b>A</b>	The value of the expression ( <code>qq - rr</code> ) is -1	<input type="radio"/>
<b>B</b>	The statement <code>printf("%d", ++ss);</code> will print "9" (without quotes)	<input type="radio"/>
<b>C</b>	Both <b>A</b> and <b>B</b> are correct	<input type="radio"/>
<b>D</b>	None of the above is correct	<input type="radio"/>

**2.4** What will happen when we try to execute the program given on the right?

<b>A</b>	Compilation error	<input type="radio"/>
<b>B</b>	Segmentation fault	<input type="radio"/>
<b>C</b>	Output will be 01235	<input type="radio"/>
<b>D</b>	Output will be 01236	<input type="radio"/>

```
#include <stdio.h>
int main(){
    int i, arr[5] = {0, 1, 2, 3, 4};
    arr[arr[4]++]++;
    for(i = 0; i < 5; i++)
        printf("%d", arr[i]);
    return 0;
}
```

**2.5** Which of the following case declarations would be illegal in a switch case statement?

<b>A</b>	<code>case 6: printf("The case is 6");</code>	<input type="radio"/>
<b>B</b>	<code>case 'a': printf("The case is a"); break;</code>	<input type="radio"/>
<b>C</b>	<code>case (float)7: printf("The case is 7.0"); break;</code>	<input type="radio"/>
<b>D</b>	<code>case (char)10: printf("The case is %c", 10);</code>	<input type="radio"/>

**2.6** Which of the following expressions would evaluate to true according to Mr C?

<b>A</b>	(1/6)	<input type="radio"/>
<b>B</b>	(1.99999 - 2)	<input type="radio"/>
<b>C</b>	(3 / 6 * 2)	<input type="radio"/>
<b>D</b>	(int)('a' - 'd' + 3.0)	<input type="radio"/>

ESC 101: Fundamentals of Computing				Mid-sem Exam (22 Sep 2018)		
Name						75 marks
Roll No		Dept.		Section		Page 3 of 8

**Q3.** Fill in the circles next to ALL CORRECT options (many may be correct). (5x3=15marks)

**3.1** Which of the following are valid variable declarations?

<b>A</b>	<code>int var_1 = 1;</code>	<input type="radio"/>
<b>B</b>	<code>int 1_var = 2;</code>	<input type="radio"/>
<b>C</b>	<code>int _ = 3;</code>	<input type="radio"/>
<b>D</b>	<code>int _4 = 4.0;</code>	<input type="radio"/>

**3.2** Which of the following expressions evaluate to value 10?

<b>A</b>	<code>3 * 6 - 9 / 2 - 9 / 2</code>	<input type="radio"/>
<b>B</b>	<code>-3 / 4 * 6 % 2 + 5 * 2 % 6 + 13 / 2</code>	<input type="radio"/>
<b>C</b>	<code>6 * 7 / 4 + 2 * 2</code>	<input type="radio"/>
<b>D</b>	<code>5 * 8 % 3 * 9 - 9 / 3</code>	<input type="radio"/>

**3.3** Which of the following are valid statements? Assume `a = b = c = 0` are **int** variables.

<b>A</b>	<code>a = 3 * 6 + 9 / 2.0;</code>	<input type="radio"/>
<b>B</b>	<code>int d = (5 == b);</code>	<input type="radio"/>
<b>C</b>	<code>c = a + 5 = b;</code>	<input type="radio"/>
<b>D</b>	<code>a *= b;</code>	<input type="radio"/>

**3.4** Mr. C wrote a program to input  $n \leq 20$  integers ( $n$  is itself an input) and print their sum. However, the code has logical errors. Which of the following combinations of line numbers are such that the code would work correctly on all compilers, if corrections are made to those lines numbers?

<b>A</b>	Lines 5 and 7	<input type="radio"/>
<b>B</b>	Lines 4 and 9	<input type="radio"/>
<b>C</b>	Lines 6 and 8	<input type="radio"/>
<b>D</b>	No lines need correction	<input type="radio"/>

```

1  #include<stdio.h>
2  int main(){
3      int a[20], sum = 0, n, i;
4      scanf("%d", &n);
5      for(i = 1; i <= n; i++)
6          scanf("%d",&a[i]);
7      for(i = 1; i <= n; i++)
8          sum += a[i];
9      printf("%d\n",sum);
10     return 0;
11 }
```

**3.5** Which of the following are valid relational/logical expressions that can be used, for example, in an if-else statement? Assume for this sub-question that `p = q = r = 1` are **int** variables

<b>A</b>	<code>(p / 6) &amp;&amp; (q &lt; 5)</code>	<input type="radio"/>
<b>B</b>	<code>(r * 3)    3</code>	<input type="radio"/>
<b>C</b>	<code>(p =&lt; r)    (q &gt;= 4)</code>	<input type="radio"/>
<b>D</b>	<code>-25</code>	<input type="radio"/>

**Q4** In the space provided, write down the output of the program when given the input indicated. Getting every output correctly carries equal weightage. **(3+4+4+4=15marks)**

**4.1** Write down the output for the given input in the space provided

INPUT	OUTPUT
8	
7	
6	

```
#include <stdio.h>
int main(){
    int n;
    scanf("%d",&n);
    do{
        if(n % 5 == 1){
            printf("%d",n % 5 + 2);
            continue;
        }else if(n % 5 == 2){
            printf("%d",n % 5 + 2);
        }else if(n % 5 == 3){
            printf("%d",n % 5);
            break;
        }
        n++;
    }while(n%5 <= 3 && n%5 >= 2);
    return 0;
}
```

**4.2** Note that the input to this program is given in two separate lines. Also assume that an empty string (i.e. where the first character itself is the NULL character) will never be given as input.

Take very good care about the case of output

INPUT	OUTPUT
thisquestion isnoteasy	
mac apple	
abigsentence isambiguous	
thiscodedoes notseemcorrect	

```
#include <stdio.h>
int main(){
    int i, j = 0, k = 0;
    int ar1[50], ar2[50];
    char ch, c1[50], c2[50];
    for(i = 0; i < 50; i++)
        ar1[i] = ar2[i] = 0;
    scanf("%s", c1);
    scanf("%s", c2);
    do{
        ch = c1[j++];
        ar1[ch - 'a']++;
    }while(ch != '\0');
    do{
        ch = c2[k++];
        ar2[ch - 'a']++;
    }while(ch != '\0');
    if(ar1[0]==ar2[0]){
        if(ar1[1]==ar2[1])
            printf("TRUE\n");
    }else
        printf("FALSE\n");
    return 0;
}
```

**4.3** Be careful that the output is in two different lines in this question. Assume that the input will never be such that segmentation faults occur.

INPUT	OUTPUT
wow	
hello world	
break	
sizzling	

```
#include <stdio.h>
int main(){
    char str[50];
    scanf("%s", str);
    int i, j, k = 0;
    for(i = 0; str[i] != '\0';i++){
        for(j = 0; j < i; j++){
            if(str[i] == str[j]){
                str[i] = '*';
                k--;
                break;
            }
            k++;
        }
        for(i = 0; i < k; i++)
            printf("%c", str[i]);
        printf("\n");
        printf("%s", str);
        return 0;
    }
```

**4.4** Please note that the output in this question may contain multiple numbers separated by a space. There might be a dangling space at the end of the output. Do not worry about showing the dangling space. We know it is there.

IN P U T	OUTPUT
10	
15	
30	
50	

```
#include <stdio.h>
int main(){
    int i, n, k = 2, alt = 1, flag;
    scanf("%d",&n);
    do{
        flag = 1;
        for(i = 2; i < k; i++){
            if (k % i == 0){
                flag = 0;
                break;
            }
        }
        if(flag){
            if (alt){
                printf("%d ",k);
                alt = 0;
            }
            else alt = 1;
        }
        k++;
    }while(k < n);
    return 0;
}
```

**Q5** In the following questions, you are either given incomplete code but with some hints inside comments to complete the code or else given complete but buggy code. Fill in the blanks neatly with code so that the program ends up doing what is specified in the question. If you need to indicate a space, leave a small gap (**don't write a dot like Prutor**). (3+4+8=15marks)

**5.1** The following program is supposed to read an English sentence with less than or equal to 99 characters, and then convert all lower case English alphabets in the sentence to upper case and vice versa. We promise the sentence will be given in a single line i.e. the sentence won't contain the newline ' \n ' character. Fill in the blanks using directions provided in the comments.

**Example:**

**INPUT:** The Month is SePtEmber

**OUTPUT:** tHE mONTH IS sEpTeMBER

```
#include <stdio.h>
int main(){
    char str[100];
    // Read the sentence from input as a string

    _____

    int i;
    for(i=0; i<100; i++)
        // Check if character at index i is in UPPER case
        if(_____)
            // Convert it into lower case.

            str[i] = _____ ;
            // Check if the character is in lower case
        else if(_____)
            // Convert it into UPPER case.

            str[i] = _____ ;
            // Otherwise it is not an alphabet.
            // Check if this is end of the string.
        else if(_____)
            break;
    printf("%s",str);
    return 0;
}
```

a. If the last digit is 0, Tom's turn is over

c. If the last digit is strictly greater than 6, Tom gets 0 runs but keeps playing

As input, you will be given two sequences of digits (separated by space) on two different lines. The first sequence will represent last digits in Tom's turn and the second sequence will represent last digits in Jerry's turn. We promise that both lines of input will contain exactly a single 0 will signal the end of that sequence, as well as that neither sequence will contain negative numbers or non-int numbers.

The following program was written to tell us who won the match (or if it was a draw). However, there are errors in the program. Point out and correct all types of errors (compilation, runtime, logical). Frivolous and unnecessary corrections may receive negative marks.

1	#include <stdio.h>	Line No	Corrected Code
2	int main(){		
3	int num;		
4	int tScore = 0;		
5	int jScore = 1;		
6	do{		
7	scanf("%d", &num);		
8	if(num == 0) break;		
9	if(num <= 6)		
10	jScore += num;		
11	}while(1)		
12	do{		
13	scanf("%d", &num);		
14	if(num <= 0) break;		
15	if(num <= 6)		
16	tScore += num;		
17	}while(1)		
18	if(tScore > jScore)		
19	printf("Tom won");		
20	if(tScore < jScore)		
21	printf("Jerry won");		
22	if(tScore == jScore)		
23	printf(Draw);		
24	return 0;		
25	}		
DO NOT SUGGEST CORRECTIONS TO MORE THAN 10 LINES			

**5.3** The following program was written to check whether a given word is a palindrome or not. A word is called a palindrome if it appears the same whether read forwards or backwards. For example the word "madam" (without quotes) or the exclamation "heh" (without quotes). In comparison, the word "sir" (without quotes) is not a palindrome. The following program was written to check if a given word is a palindrome or not but has errors.

First write down what output will this **incorrect program give on the given inputs**. Then correct the program by pointing out line numbers and corrections to those line numbers. Frivolous and unnecessary corrections may receive negative marks.

```
1  #include <stdio.h>
2  int main(){
3      char str[100];
4      scanf("%s", str);
5      int p, sta = 1, fin;
6
7      for(p = 0; p < 100; p++)
8          if(str[p] == '\n')
9              break;
10
11     fin = p;
12     int isPalindrome = 0;
13
14     while(sta < fin){
15         if(str[sta] != str[fin]){
16             isPalindrome = 1;
17             break;
18         }
19         sta++;
20         fin--;
21     }
22
23     if(isPalindrome)
24         printf("YES");
25     else
26         printf("NO");
27     return 0;
28 }
```

INPUT		OUTPUT
Alice		
WoW		
Mom		
Line No	Corrected Code	
DO NOT SUGGEST CORRECTIONS TO MORE THAN 7 LINES		