

ESC 101N: Fundamentals of Computing

(2010-11-IIInd Semester)

Mid Sem II Examination

3.30-4.30PM, Monday 7th March, 2011

Instructions

1. Write your name, roll number and section below and also in the space provided at the top of every subsequent page.
2. Read the instructions, and don't turn over until the second bell rings.
3. Write the solution in the space provided under each question. Don't use pencil.
4. Deposit cell phone (if you carry one) at the desk. Collect it after the exam.
5. Please submit the paper to one of the invigilators after the exam.
6. Paper contains 5 question in 7 pages.

Name:
Roll No:
Section:
Signature:

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

1. (a) Consider a C program which has following initialization of 2D array.

```
#include <stdio.h>
int main() {
int i, j;
    int a[][5] = {[0][0] = 8, [1][3] = 9, [0][4] = 7, [2][0] = 6,
                  [1][4] = 6, [2][2] = 6, [3][0] = 6, [2][4] = 6,
                  [1][2] = 7, [3][3] = 9
                };
    // print statements for printing a
}
```

Assume that correct print statements have been provided for printing each row of the array (starting with first row first element) in a separate line with each element separated by a comma. What will be the values printed for each row?[4]

Solution (set A/set B):

The values printed for are:

8,0,0,0,7/ 0,8,7,0,0 [1 Mark]

0,0,7,9,6/ 0,0,0,7,0 [1 Mark]

6,0,6,0,6/ 0,6,6,6,6 [1 Mark]

6,0,0,9,0/ 0,6,0,0,9 [1 Mark]

- (b) Go through the C program given below. Determine the values of a, b, c, d that get printed by taking care of the order in which the print statements are executed. [6]

```
#include <stdio.h>

int b = 5;
int c = 15;

void f() {
    int b, d = -5, a = 0;
    c = c-d;
    b = c + d;
    a = b;
    printf("a = %d, c = %d\n", a, c);
}

int d = 25;

void g() {
    int c;
```

```

    c = 10;
    c = b - 5;
    {
        int c;
        c = b + 5;
        b = b - c;
        d = d + 10;
        printf("b = %d, d = %d\n", b, d);
    }
    printf("b = %d, c = %d\n", b, c);
}

int main() {
    int a = 10, d = 5;
    f();
    g();
    printf("a = %d, d = %d\n", a, d);
}

```

Solution (set A / set B):

a = 15, c = 20 / a = -5, c = 10	[both correct 1.5/single .5]
b = -5, d = 35 / b = -5, d = 20	[both correct 1.5/single .5]
b = -5, c = 0 / b = -5, c = 5	[both correct 1.5/single .5]
a = 10, d = 5 / a = -5, d = 15	[both correct 1.5/single .5]

2. (a) Give 8-bit 1's complement representation of -45 and 20. [2]
- (b) Give 8-bit 2's complement representation of -100. [2]
- (c) Give the representation of -0.75 using a 9-bit floating point representation which has the following format: [6]
- i. 1 bit sign
 - ii. 4 bit exponent (in excess 8 representation)
 - iii. 4 bit mantissa (with implied 1.0)

Solution (set A / Set B):

(a) 11010010, 00010100 / 11001010, 00010111 [2 Marks]

(b) 10011100 / 10011110 [2 Marks]

(c) Normalized representation:

For set A

$(-0.75)_{10} = -(.5 + .25)_{10} = (0.11)_2 \times 2^0 = (1.1 \times 2^{-1})_2$. with implied 1 [2 Mark]

Biased exponent (without implied 1): $(0 + 8)_{10} = 8_{10} = (1000)_2$. [2 Mark]

So the representation without implied 1 is: 0-1000-1100.

The representation with implied 1 is: 1-0111-1000 [2 Mark]

For set B

$(-0.5675)_{10} = -(.5 + .0625)_{10} = (0.1001)_2 \times 2^0 = (1.001 \times 2^{-1})_2$. with implied 1 [2 Mark]

Biased exponent (without implied 1): $(0 + 8)_{10} = 8_{10} = (1000)_2$. [2 Mark]

So the representation without implied 1 is: 0-1000-1001

The representation with implied 1 is: 1-0111-0010 [2 Mark]

3. Following program takes two numbers x and y as input and prints the number of prime numbers in the range [x, y], both inclusive (either x or y or both may be primes). The code segments marked ?1? through ?5? are not clear. Provide the correct substitutes (use of comma operator is not permitted) for those segments so that the program works as indicated. [2 × 5]

```
#include <stdio.h>
int isPrime(int n) {
    int divisor;
    if (n == 2)
        return 1;
    if (n == 1 || n % 2 == 0)
        return 0;
    for (divisor = 3; ?1?; divisor = divisor + 2)
        if (n % divisor == 0)
            return 0;
    return 1;
}
```

```
int countPrimes(int a, int b) {
```

```

    if (??)
        return 0;
    if (??)
        return ?4?;
    else
        return countPrimes(a, b-1);
}

int main() {
int x, y;
    printf("Enter first number: ");
    scanf("%d", &x);
    printf("Enter second number: ");
    scanf("%d", &y);
    printf("Primes between %d and %d = %d\n", x, y, ?5?);
}

```

Solution (same for both sets):

```

?1?: divisor * divisor <= n
?2?: a > b
?3?: isPrime(b)
?4?: countPrime(a, b-1) + 1
?5?: countPrime(x, y)

```

4. Carefully understand the following C function.

```

#include <stdio.h>
void shuffle(char w[], int left, int right) {
    char c;
    if (left < right) {
        c = w[left];
        w[left] = w[right];
        w[right] = c;
        shuffle(w, left+1, right-1);
    }
    return;
}

int main() {
    char w[] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'};
    char x[] = {'n', 'a', 'c', ' ', 'y', 'o', 'u', ' ', 'e', 't', 'i', 'r',
                'w', ' ', 'C', ' ', 'e', 'd', 'o', 'c'};
}

```

```

    int n = sizeof(w)/sizeof(w[0]);
    int m = sizeof(x)/sizeof(x[0]);
    int i, j;

    shuffle(w, 2, 4);
    for (i = 0; i < n; i++)
L1     printf("%c", w[i]);
L2 printf("\n");
    shuffle(w, 0, 4);
    for (i = 0; i < n; i++)
L3     printf("%c", w[i]);
L4 printf("\n");

    shuffle(x, 0, 2);
    for (i = 0; i < m; i++)
L5     printf("%c", x[i]);
L6 printf("\n");
    shuffle(x, 8, 12);
    for (i = 0; i < m; i++)
L7     printf("%c", x[i]);
L8 printf("\n");
    shuffle(x, 16, 19);
    for (i = 0; i < m; i++)
L9     printf("%c", x[i]);
L10 printf("\n");
}

```

Now determine the output of the program taking care of the order of print statements in the program. [2×5]

Solution (set A/set B):

L1-L2; abedcfgh / hfdefcba	[2 Marks]
L3-L4: cdebafgh / fedghcba	[2 Marks]
L5-L6: can you etirw C edoc / you can etirw C edoc	[2 Marks]
L7-L8: can you write C edoc / you can write C edoc	[2 Marks]
L9-L10: can you write C code / you can write C code	[2 Marks]

5. Find the output of the following program. [10]

```

#include<stdio.h>

void g(int n);

```

```
void f(int n) {
    if(n==0)
        return;
    g(n-1);
    printf("-%d ", n);
    f(n-1);
}

void g(int n) {
    if(n==0)
        return;
    f(n-1);
    printf("%d ", n);
    g(n-1);
}

int main() {
    printf("First call: ")
    f(2);
    printf("\n");
    printf("Second call: ")
    f(3);
    printf("\n");
}
```

Solution (set A / Set B):

First call: 1 -2 -1 / -1 2 1 -3 1 -2 -1

[3 Marks]

Second call: -1 2 1 -3 1 -2 -1 / 1 -2 -1

[7 Marks]

Space for Rough Work

Roll:

Name:

Section:

Space for Rough Work