

1	2	3	4	5	6	7	8	9	Total

Indian Institute of
Technology Mandi**IC150: Computation...**
14th June 2012, 9:00-12:00

Final Exam

Answer all questions. **No calculators or cellphones.**

Maximum marks: 70

- 0) One lesson learnt in this course that I will remember for the rest of my life is: [½]
Popular answers: punctuality, discipline, being regular in class and studies, taking notes, C programming, practice makes perfect

Other answers: structures, how to learn, how to teach, to have a passion, programming is an art, pointers, problem solving methods, Scilab, not to sleep in class, work hard, many ways to solve a given problem, Moodle, programming is fun, read newspaper daily.

Note: other answers may be possible.

- 1) Fill in the blanks: [5]

- An else clause is often used after an `if` clause.
- A C expression involving the operator '=', eg. `x = y = 5`, is evaluated from right to left
- The decimal value of 2^{16} is 65536
- The number of bytes of memory allocated to a variable depends on its type
- A standard C function to convert a string to an integer is atoi() or sscanf()
- The standard C function to clear the I/O buffers is fflush()
- The standard C function to allocate memory dynamically is malloc() or calloc()
- The time complexity of binary search in an array of N integers is $O(\log N)$
- The function prototype of the main function in a C program is normally written as
`int main(int argc, char * argv[])`
or `char ** argv`

- 2) Answer briefly: [12]

- Explain the difference between the file open modes "w" and "w+".
"w" – open an existing or new file for writing.
"w+" -- open an existing or new file for reading and writing.
- Explain the possible return values of `strcmp(s, t)`.
 $<0 \rightarrow s < t$ (lexicographically)
 $0 \rightarrow s == t$
 $>0 \rightarrow s > t$
- Write C statements to declare an integer variable containing the value 17, and a pointer variable that points to this integer.
`int n = 17;`
`int *p = &n;` or equivalent
- Is the Bisection Method guaranteed to converge? Explain.
Yes. In every iteration, the interval is halved.
- List three sources of error in numerical analysis.
- Round off error
- Finite precision of representation
- approx. eval. of functions, e.g. $\pi = 22/7 \approx 3.14159$
- Explain either *least squares fit* or *minimax fit* (choose any one).
Used in regression analysis (curve fitting where the curve need not pass through the

data points).

(1) Minimize mean square error i.e. sum of the square of deviation of each data point from the function value at that x

OR (2) Minimize maximum error, i.e. $\text{Max}_k |f(x_k) - y_k|$

g) Explain *space-time tradeoff* with an example.

For solving a problem, different algorithms may exist. One is faster but uses more memory, the other is slower but uses less memory. Eg. radix vs comparison sorting. OR recursive Fibonacci vs. Saving intermediate Fibonacci values rather than recomputing them.

h) Why do C and Linux provide the abstraction of an *I/O stream*?

I/o devices are electro-mechanical with very diverse characteristics. To make it easier for the programmer, a uniform abstraction "stream" is provided for all devices.

3) (a) Write Scilab code to create the polynomial $3x^3 - 4x^2 + 5x - 12$ and store it in the variable p . [2]

```
p = poly([-12 5 -4 3], 'x', "coeff");
```

(b) Give 3 important advantages of Scilab over C. [3]

1. Interpreted => immediate execution
2. Variables need not be declared
3. Arrays are basic data types, can be used in expressions
4. Many builtin functions for stats, plotting, various branches of engineering, etc.

4) Do the indicated conversions [4½]

a) $(603)_{10}$ to binary

1001011011_2

b) $(110101)_2$ to unsigned decimal

53_{10}

c) $(110101110101101)_2$ to hexadecimal

$6bad_{16}$

5) (a) Create a struct `Ticket` to hold details of a travel booking. For each booking the details are name (max 30 characters), 5-digit train number (eg, 12353) and price of the ticket in rupees. [3]

```
struct Ticket {
    char name[31];
    int trainNum;
    int/float price;
}
```

(b) Fill in the code for the function `TotalPrice()` that returns the total price of the `num` tickets in group. [2]

```
int TotalPrice(struct Ticket group[], int num)
{
    int p=0;
    for (i=0; i<num; i++)
        p += group[i].price;

    return p;
}
```

6) (a) Write a complete C program `recho.c` that outputs its command-line arguments in reverse order. I.e., if we type at the terminal prompt : [5]

```
$ recho one two three
```

the output will be:

```
three two one
#include <stdio.h>
int main(int argc, char *argv[])
{
    int i;
    for (i=argc; i>0; i--)
        printf("%s", argv[i]);
    printf("\n");
    return 0;
}
```

(b) Specify two test cases for your program, one "normal" and the other a "boundary" test case (different from the example given above).

\$ recho (bdy case, no output)
\$ recho a bcd a grs (normal case)
grs a bcd a

7) Given 3 sides of a triangle, a , b and c as real numbers. [4]

a) Write a snippet of C code that sets the variable `isRt` to 1 if the 3 numbers represent a right-angled triangle or 0 otherwise.

```

1. a, b, c
2. check if  $a^2 = b^2 + c^2$ 
    if (b>a) Swap (&a,&b);
    if (c>a) Swap (&a, &c);
    if (    =+c*c) isRt=1
    else isRt=0;

void Swap(float&*p, float*q)
{
    float tmp;
    tmp = p;
    p = q;
    q = tmp;
}

```

Aliter:

```

if((a<b+c)&&(b<c+a)&&(c<a+b))
    if((a*a == b*b + c*c) || (b*b == c*c + a*a) ||
        (c*c == a*a + b*b))
        isRt = 1;
    else isRt=0;
else printf("Not a triangle\n");

```

b) Under what conditions (specific values of a , b and c) is your code likely to fail to give the correct result?

Fractional values may fail due to limited precision
eg. a = 0.0003, b = 0.0004, c = 0.0005

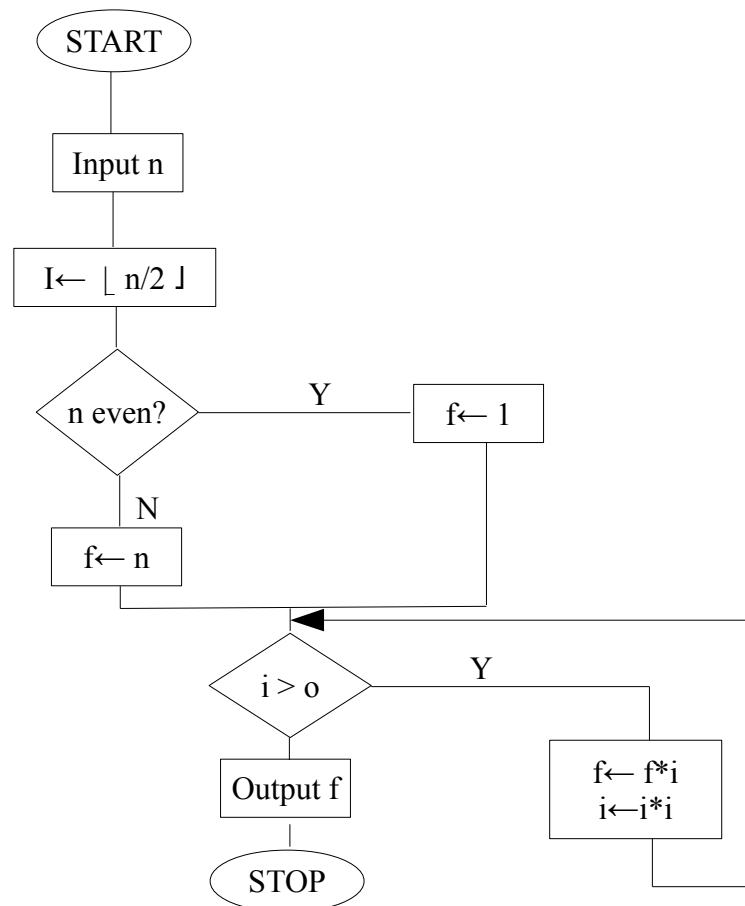
8) Given the function $f(n) = (n/2)!$ For even n [5]
 $= n \times f(n-1)$ For odd n

a) What is the value of $f(3)$ and $f(8)$?

$$f(3) = 3 \times f(2) = 3 \times 1! = 3$$

$$f(8) = (4)! = 24$$

Draw a neat flowchart to iteratively compute the value of $f(n)$ for any positive n .



9) After execution of the following program, are the elements in array6 reversed? Justify your answer in short. Outline the **minimum necessary changes** to ensure that the elements in array6 are reversed at the end. List the changes clearly. [4]

```

#include <stdio.h>
int main(void)
{
    int array6[6] = {1,2,3,4,5,6};
    int i, tmp;

    for(i=0; i=5; i++)
    {
        tmp      = array6[i];
        array6[i] = array6[5-i];
        array6[5-i] = tmp;
    }
}
  
```

1. As the dimension of array6 is wrong, the array6[5] may overwrite tmp.
2. Assuming this does not happen, in the 1st half of the for

loop, elements of array6 get reversed, in the 2nd half they again get reversed to result in the original order.

Corrections:

```
1. int array6[6] = {1,2,3,4,5,6};
2. int i, tmp;
3. for (i=0; i <= 2; i++)
```

- 10) The file input.dat contains one integer on each line. Write a C programme that reads input.dat and writes all the non-negative integers only to the file output.dat, one number per line. Assume that both files exist and are readable/writable respectively. [5]

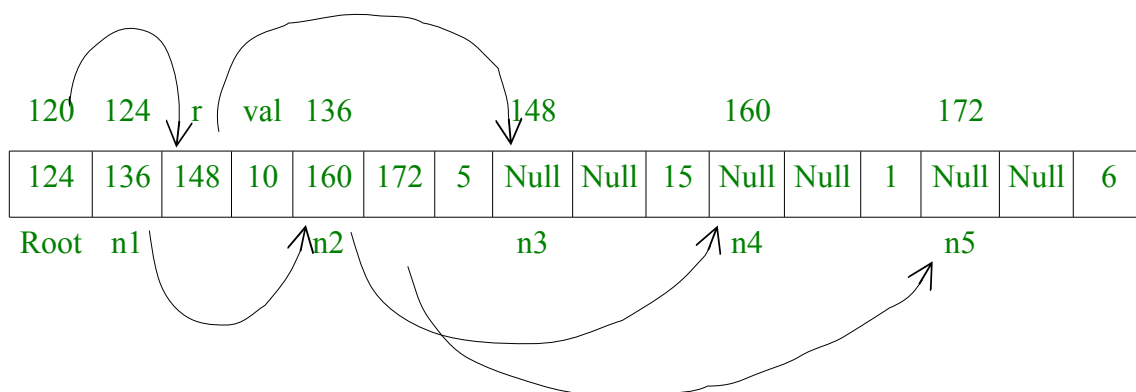
```
#include <stdio.h>
int main()
{
    FILE * inf*, *outf;
    int n;

    inf=fopen(.....); outf = fopen (.....);
    while (!feof(inf)) {
        fscanf (inf, "%d" &n);
        if (n>=0) fprintf(outf,"%d\n", n);
    }
    fclose(inf); fclose(outf);
}
```

- 11) A binary tree is a dynamic data structure that is similar to a linked list. Consider the code below. Draw a neat picture of the memory allocated, showing the fields of each structure and their numeric values at the end of execution. Next draw arrows indicate the pointers. Assume that an int and a pointer occupy 4 bytes each, and that the variables are allocated to contiguous memory locations starting at location 120. [4]

```
struct node {
    struct node *left, *right;
    int val;
} *root, n1, n2, n3, n4, n5;
```

```
root = &n1;
n1.val = 10; n1.left = &n2; n1.right = &n3;
n2.val = 5; n2.left = &n4; n2.right = &n5;
n3.val = 15; n3.left = NULL; n3.right = NULL;
n4.val = 1; n4.left = NULL; n4.right = NULL;
n5.val = 6; n5.left = NULL; n5.right = NULL;
```



- 12) Suppose $p(x) = 3x^3 - 4x^2 + 5x - 12$. [5]
 (a) Use Horner's method to find the value of $p(4)$. Show your calculations clearly.

$$\begin{aligned} & ((3x-4)x+5)x-12 \\ = & ((3 \times 4 - 4)4 + 5)4 - 12 \\ = & ((32 + 5)4 - 12) = 136 \end{aligned}$$

- (b) Given the initial interval $[x_0, x_1] = [0, 2]$, what is the new interval after one iteration of the Regula-falsi method?

$$\begin{aligned} p(0) &= -12 \\ p(2) &= 3 \times 8 - 4 \times 4 + 5 \times 2 - 12 = 6 \\ \Rightarrow \\ p\left(\frac{4}{3}\right) &= \frac{3 \times 4^{-3}}{3^{-3}} - \frac{4 \times 4^{-2}}{3^{-2}} + \frac{5 \times 4}{3} - 12 \\ &= (64 - 64) / 9 + 20 / 3 - 12 = -5.333 < 0 \\ \Rightarrow \text{new interval is } [1.3333, 2] \end{aligned}$$

- 13) Consider the function `int Vpattern(int a[], int aSize)` which takes as input an array of distinct integers called `a` and an integer `aSize` that gives the number of elements present in `a`. The function returns 1 if the integers in the array are such that the first half is a decreasing sequence and the second half is an increasing sequence, otherwise it returns 0. Assume that the array size is even. [6]

- (a) Give an example array of size 8 integers that satisfies the above condition.

[12 10 8 6 1 10 20 40]

- (b) Give an example array of size 8 integers that does not satisfy the above condition.

[12 10 20 5 2 4 3 1]

- (c) Explain the algorithm you would use to solve the problem (use pseudo-code).

1. For the left half, check if each num is > its right neighbour.
2. For the left half, check if each num is < its right neighbour.
3. If both above are true, return 1, else return 0.

Refinement:

0. Let array index range from 0..aSize-1

1. For $i = 0$ to $aSize/2 - 2$ do

1.1. if $a[i] \leq a[i+1]$ then

1.1.1 return False

1.2. end For

2. For $i = aSize/2$ to $aSize - 2$ do

1.1. if $a[i] \leq a[i+1]$ then

1.1.1 return False

1.2. end For

3. Return True