



INDIAN INSTITUTE OF TECHNOLOGY
KHARAGPUR

Stamp / Signature of the Invigilator

EXAMINATION (Mid Semester)

SEMESTER (Spring)

Roll Number

Section

Name

Subject Number

C S 1 1 0 0 1

Subject Name

Programming and Data Structures

Department / Center of the Student

Additional sheets

Instructions and Guidelines to Students Appearing in the Examination

1. Ensure that you have occupied the seat as per the examination schedule.
2. Ensure that you do not have a mobile phone or a similar gadget with you even in switched off mode. Note that loose papers, notes, books should not be in your possession, even if those are irrelevant to the paper you are writing.
3. Data book, codes or any other materials are allowed only under the instruction of the paper-setter.
4. Use of instrument box, pencil box and non-programmable calculator is allowed during the examination. However, exchange of these items is not permitted.
5. Additional sheets, graph papers and relevant tables will be provided on request.
6. Write on both sides of the answer script and do not tear off any page. Report to the invigilator if the answer script has torn page(s).
7. Show the admit card / identity card whenever asked for by the invigilator. It is your responsibility to ensure that your attendance is recorded by the invigilator.
8. You may leave the examination hall for wash room or for drinking water, but not before one hour after the commencement of the examination. Record your absence from the examination hall in the register provided. Smoking and consumption of any kind of beverages is not allowed inside the examination hall.
9. After the completion of the examination, do not leave the seat until the invigilator collects the answer script.
10. During the examination, either inside the examination hall or outside the examination hall, gathering information from any kind of sources or any such attempts, exchange or helping in exchange of information with others or any such attempts will be treated as adopting 'unfair means'. Do not adopt 'unfair means' and do not indulge in unseemly behavior as well.

Violation of any of the instructions may lead to disciplinary action.

Signature of the Student

To be filled in by the examiner

Question Number

1

2

3

4

5

6

7

8

9

10

Total

Marks Obtained

Marks obtained (in words)

Signature of the Examiner

Signature of the Scrutineer

Instructions: Answer all six questions. Total marks = 60. Time = 2hrs. Write your answer only in the space provided. Use any other space for rough work. The question paper has total 8 pages.

Rough Work

1. In questions (i-v) tick the correct choices. More than one choice may be correct.

[5 x 2]

i. While writing a C program we directly interact with:

A. Compiler B. Operating system **C. Editor** D. Web browser E. Linker

ii. Which of the following is true for the program below?

```
#include <stdio.h>
void main () {
int num1 = 5, num2 = 10;
if(num1 < num2){
    printf("num1 less num2");
    else
    printf("num1 not less num2");
}
}
```

- A. Will compile and output "num1 less num2"
- B. Will compile and output "num1 not less num2"
- C. Will not compile successfully**
- D. None of the above

iii. Elnino Pizza has found out that IIT Kgp students have finished their mid semester examination. They have announced a special discount of 10% on purchases priced strictly greater than Rs. 500. Purchases having price less than or equal to Rs. 500 have no discount. Assume that a float variable 'pizza_price' stores the price of the pizza. Which of the following statements will correctly calculate the price after discount?

- A. `pizza_price -= (pizza_price > 500 ? pizza_price * 1.1 : pizza_price);`
- B. `pizza_price -= (pizza_price > 500 ? pizza_price * 0.1 : pizza_price);`
- C. `pizza_price -= (pizza_price > 500 ? pizza_price * 0.9 : pizza_price);`
- D. `pizza_price -= (pizza_price > 500 ? pizza_price * 0.1 : 0);`**
- E. None of the above

iv. A student has written the program shown below for doing some mysterious computation. Suppose we compile and run this program. Which of the following is true for the number of "!"s printed by the program?

- A. For some values of 'm' and 'n', exactly one "!" is printed.
- B. For all values of 'm' and 'n', no "!" is printed.**
- C. For all values of 'm' and 'n', the number of "!"s printed is either 1 or a positive even number.
- D. For all values of 'm' and 'n', the number of "!"s printed is a positive odd number.
- E. None of the above

```
#include <stdio.h>
void main() {
int m, n, temp;
printf("Give integers m and n: ");
scanf("%d%d", &m, &n);
do {
    if (m < n) n = n - m;
    if (n < m) {temp = m; m = n; n = temp;}
    if (m > n) printf("!");
} while (m < n);
}
```

v. What is the output printed by the following program?

```
#include<stdio.h>
void main(){
int A[]={0,1,2,3,4,5,6,7};
int n=8, step = 2, i, j, k, l, temp;
for(i=0;i<n-step;i++){
    for(j=i;j<i+step;j++){
        temp = A[j];
        A[j] = A[j+1];
        A[j+1] = temp;}
step = (step*2)- 1;}
for(i=0; i<n; i++)printf("%d ",A[i]);
}
```

- A. 1 2 0 3 4 5 6 7
- B. 1 0 3 4 2 5 6 7
- C. 1 0 2 4 6 5 7 3
- D. 1 0 4 2 5 6 7 3**
- E. None of the above

2. Given an integer nn as input, the following code prints the patterns shown below using 'X' and the space character. (You only need to handle odd integers greater than 4.) Fill in the missing pieces in this code: **[10]**

Input=7

```
XXX XXX
XX  XX
X   X
   X
X   X
XX  XX
XXX XXX
```

Input=9

```
XXXX XXXX
XXX  XXX
XX   XX
X    X
   X
X    X
XX   XX
XXX  XXX
XXXX XXXX
```

Input=11

```
XXXXX XXXXX
XXXX  XXXX
XXX   XXX
XX    XX
X     X
   X
X     X
XX    XX
XXX   XXX
XXXX  XXXX
XXXXX XXXXX
```

```
int nn; scanf("%d",&nn);
int half = nn/2;
int isNWCorner, isNECorner, isSWCorner, isSECorner ;
int isCorner, isCenter ;
int row, col ;
for (row = 0; row < nn; ++row) {
for (col = 0; col < nn; ++col) {

isNWCorner = row + col < half ;

isNECorner = col - row > half ;

isSWCorner = row - col > half;

isSECorner = row + col > nn + half - 1 ;

isCorner = isNWCorner || isNECorner || isSWCorner || isSECorner;
isCenter = (row == half) && (col == half);
printf("%c",(isCorner || isCenter)? 'X' : ' ');
}
printf(" \n ") ;
}
```

3. The royal palace in Dholakpur employs $G (< 100)$ guards numbered $0, 1, \dots, G-1$. Assume each guard $g: 0 \leq g < G$ reports at midnight and leaves the next midnight. During this 24-hour period, a guard can take any number of breaks at any time s/he chooses, but has to enter into a register exactly when s/he left the palace and returned after the break. Each line of the register records an event and has three columns:

- The guard number g (`int guard[]`)
- The time of the event e (`float time[]`).
- What happened: arrival or departure (`int what[]`, is 1 if arrival and 0 if departure)

The register records r are stored in three arrays as defined above. These arrays are sorted on time. Total R records are present in the register.

i. A famous diamond was stolen at time 4.08 (just after 4am). The Chief Inspector wants a list of all guards who were on duty at that time. Complete the following code which does this. Assume value of G is available. [5]

```
int r = 0, g, onduty[100];
float T = 4.08;

for(g = 0; g < G; g++) onduty[g] = 1;

while(time[r] <= T){

onduty[guard[r]] = what[r];
r++;
}
for(g = 0; g < G; g++)

if(onduty[g]) printf("Guard %d was on duty.", g);
```

ii. The Chief Inspector suspects that security in the palace has gotten too lax. He wants to find out the total duration through the last 24 hours when there was not a single guard on duty in the palace. Complete the following code which does this. Assume value of G , and the number of records R is available. Also assume that every guard returns to the palace after their last break before midnight. [5]

```
int r = 0, gcount = G;

float tcount = 0.0;

while(r <= R){

    if (what[r] ==1) gcount++;

    else gcount--;

    if(gcount == 0)tcount = tcount + (time[r+1] - time[r]);

    r++;

}

printf("Total duration = %f", tcount);
```

4. A polynomial $P(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1}$ is represented in sparse format using one integer and two arrays:

- `int tn` stores the number of terms in the polynomial with nonzero coefficient.
- `int powers[]` stores powers of `x` in strictly increasing order.
- `float coeffs[]` stores non-zero coefficients. Zero coefficients are not stored.

For example, $P(x) = 2x^3 + 4.3x^5$ is stored as `tn=3, powers=(0; 3; 5)` and `coeffs=(2; -1; 4.3)`.
 $P(x) = 1.3x^{15} - 4.5x^{23}$ is represented as `tn=2, powers=(15; 23)` and `coeffs=(1.3; -4.5)`.

i. Complete the code below to update "in place" `tn`, `powers` and `coeffs` to represent the derivative polynomial $P'(x) = dP/dx = \sum_{i \geq 1} i a_i x^{i-1}$ "in place" means you cannot allocate another vector of similar size. Be careful about the presence or absence of the `a0` term. **[5]**

```
int wx = 0, rx = 0;

for (; rx < tn; ++rx) {

    if (powers[rx] > 0) {

        coeffs[wx] = coeffs[rx] * powers[rx];

        powers[wx] = powers[rx] - 1;

        ++wx;
    }
}

tn = wx;
```

ii. Given a float value for `x` in the variable `x`, complete the code below to evaluate $P(x)$ and store the result in the variable `ans`. E.g., if $P(x) = 1 - x + 3x^3$ and `x = 2`, the value of `ans` should be $1 - 2 + 3 * 8 = 23$. Note that many powers of `x` may be missing, i.e., has zero coefficients. **[5]**

```
float ans = 0, x, maxVal = 1;

int maxPower = 0;

for (tx = 0; tx < tn; ++tx) {

    while (maxPower < powers[tx]) {

        maxVal *= x;

        ++maxPower;
    }

    ans += coeffs[tx] * maxVal;
}
```

5. I want to copy every alternate character of a given string `mystring` in reverse order and store them in an array `A[]`. E.g., if `mystring` is "iitkgp", then `A = {'p', 'k', 'i'}`. Complete the code below to do so.

```
char mystring[40], A[20];
scanf("%s", mystring);

l = 0 ____;

while(mystring[l]) l++; /* find the length of the string */

/* read the string in reverse order */

for (j = l-1, i=0 ; j >= 0 ; j=j-2, i++ )

A [ i ] = mystring[ j ];
```

[10]

6. A function `redundant(int list[], int n)` prints once all the duplicate (appearing multiple times) entries in an integer array.

- i. What will be the return type of the function? _____
- ii. Complete the function body below.

[1 + 3 + 2 + 4]

```
{int i, j;
for(i=0; i < n-1 ; i++){
    for (j=i+1 ; j < n ; j++)
        if (list[i] == list[j])
            printf("\n %d is duplicate in list[i]", list[i]);
}}
```

iii. What is the problem with this function?

iv. Rectify the function with minimum changes and write in the box below.

Rough Work