ID. No. Name:

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

Second Semester 2017-2018

Date: 11-May-2018
Time: 3 PM

CS F111 Computer Programming

COMPREHENSIVE EXAMINATION SOLUTIONS

Max. Marks: 75 (37.5%)
Duration: 3 hours

- The paper is divided into two parts – A which is closed-book type and B, which is open-book type.
- You should begin attempting Part-A first in the question paper itself.
- The maximum time allowed for Part-A is one hour. However, you may turn in the paper earlier.
- After turning in Part-A, you would attempt Part-B in an answer book you should collect from the invigilator.

Duration: 1 hour PART-A (Closed-book) Max. Marks: 25 (12.5%)

1. *Correct the logical errors in each of the following excerpts of code/command, if any. If you feel there is no error in the code, then write "No error" in the box provided.* **[6 x 1½ = 9]**

(i) Logic: A year is a leap year if it is divisible by 4 but not by 100, except that years divisible by 400 *are* leap years.

```
if ((year % 4 == 0 && year % 100 != 0) && year % 400 != 0)
      printf("%d is a leap year\n", year);
else
      printf("%d is not a leap year\n", year);
```

Write the corrected condition for the `if` statement here, if you feel it is erroneous:

```
if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)
```

(ii) Intent: To set the last six bits of the integer x to zero. `x | ~077;`
Write the correct statement in the box provided.

```
x = x & ~077;
```

(iii)
```
void my_strcpy(char *s, char *t)
{ /* copy t to s */
      while (*s++ = *t++)
            ;
}
```

No error.

Write the body of the function, if correction is needed. Unary operators used here associate R→L.

(iv) Intent: To append to `output.txt` details of the first 20 first-degree users whose details are found in `/etc/passwd` on the *Prithvi* server.

```
grep "^f" /etc/passwd | head -20 >> output.txt
```

```
grep "f" /etc/passwd | tail -20 > output.txt
```

(v) Intent: To enable all users to be able to view `data.txt`, but leave the other privileges untouched.

```
chperm 400 data.txt
```

`chmod a+r data.txt` or `chmod ugo+r data.txt` or `chmod +r data.txt`

(vi) Intent: To allocate dynamic memory for an array of 5 integer pointers.

```
int *arr = calloc (5, sizeof(int));
```

```
int **arr = calloc (5, sizeof(int *));
```

2. *For each of the following 6 multiple-choice questions, pick and write the most appropriate option.* **[6]**

(i) Given the definitions `struct point { int x; int y;};` and `struct rect { struct point pt1; struct point pt2; };` and the declaration `struct rect r, *rp = &r;` which of these three expressions is/are equivalent to `r.pt1.x`?

A. `rp->pt1.x` B. `(r.pt1).x` C. `(rp->pt1).x` D. All of these.

D

(ii) What is the output of the following snippet of code?

```
void fun(int *p)
{
  int q = 10;
  p = &q;
  return;
}
```

```
int main()
{
  int r = 20;
  int *p = &r;
  fun(p);
  printf("%d", *p);
  return 0;
}
```

B

A. `10` B. `20` C. Compiler-dependent D. Runtime error. E. Compiler error

(iii) Assuming variable `i` has been declared, pick the correct statement:

```
for (i = 0; i < 10 ; i++)       // (i)
for ( ; i < 10 ; i++)           // (ii)
for (i = 0;   ; i++)            // (iii)
for (i = 0; i < 10 ; )          // (iv)
for ( ; ; )                     // (v)
```

D

A. Only (i) would compile successfully.
B. Only (i) and (v) would compile successfully. Also (v) can be used as infinite loop
C. All would compile successfully but behavior of (ii), (iii) and (iv) would depend on compiler.
D. All would compile successfully.
E. None of these statements is correct.

A

(iv) `getchar()` is equivalent to which of these?
A. `getc(stdin)` B. `fgets(stdin)` C. `fscanf("%c",&ch,stdin)` D. None of these

(v) What is the output of this program?

```
int main()
{
    int a[5] = {1,2,3,4,5};
    int *ptr = (int*)(&a+1);
    printf("%d %d", *(a+1), *(ptr-1));
    return 0;
}
```

A

A. `2 5` B. Compile-time error D. Garbage values E. Runtime error

(vi) Which of the following is true of a `static` variable declared inside a function?
A. It has lifetime like a global variable because they are allocated memory in the heap.
B. It cannot be initialized by the user.
C. Its scope is same as an external variable, but lifetime same as automatic variable.
D. Every time the function is called, the static variable is allocated new memory, except when the function is called recursively.
E. None of the above statements is correct.

E

3. What is the output of the following error-free program? **[3]**

```
#include <stdio.h>
int x = -1, y = -2, z = -3;
void f1(int,int);

int main()
{
int z = 3;
x += y += ++z;
printf("%d %d %d\n",x,y,z);
{
float y = 4.0;
int x = 0;

x += z = 5 * y;
printf("%d %.2f %d\n",x,y,z);
f1(x,z);
}
printf("%d %d %d\n",x,y,z);
}

void f1 (int x, int z)
{
y = ++x + --z + (y+=2);
printf("%d %d %d\n",x,y,z);
}
```

```
1 2 4
20 4.00 20
21 44 19
1 44 20
```

4. Shown is the incomplete code to print the following pattern: **[3]**

```
    A
   BAB
  CBABC
 DCBABCD
EDCBABCDE
```

Fill in the missing parts of the code, writing your answers in the box provided.

```
int main()
{
    int i,j;
    char CH='A';
    int space=4;

    /*loop for row*/
    for(i=1; i<=5; i++)
    {
        /*put space*/
        for(j=1; j<=space; j++)
            printf(" ");
        /*first part of the row*/
        for(____________________)
            printf("%c",j);
        /*second part of the row*/
        for(_____________________)
            printf("%c",j);
        printf("\n");
        CH++;
        ____________;
    }

    return 0;
}
```

j = CH; j >= 'A'; j--

j = 'B'; j <= CH; j++

space = space – 1

5. The following program was written to check the "Endian-ness" of the computer system that is known to use 2 bytes for `short int`. The output of the program was:

```
char[0]: B
char[1]: A
```

It was thereby concluded that the Little Endian scheme was being followed. What number would have been stored into `e1.s`? Give the binary and the decimal equivalent of the number. Show the steps how you arrived at the answer. (Note: ASCII value of 'A' is 65, and that of 'B' is 66.) **[4]**

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
   typedef union {
           short s;
           char arr[2];
           } ENDIAN;
   ENDIAN e1;
   e1.s = (short) atoi(argv[1]);
   printf("char[0]: %c\n",e1.arr[0]);
   printf("char[1]: %c\n",e1.arr[1]);
}
```

The number in base 10:

16706

Binary equivalent of the number:

$$\underbrace{\texttt{01000001}}_{\text{Value of 'A' in ASCII; stored in the higher memory address}}\ \underbrace{\texttt{01000010}}_{\text{Value of 'B' in ASCII; stored in the lower memory address}}$$