Birla Institute of Technology & Science, Pilani
Second Semester 2016-2017, CS F111 Computer Programming
Mid Semester Test (Open Book)

Date: March 08, 2017 **Time: 11.00 - 12.30 PM** **Max. Marks: 78**

[*NOTE: Answer all parts of a question sequentially. Assume necessary header files are included. Also, you are not required to validate the input taken from the user by any program.*]

Q1. On **first page** of your answer sheet write (*only*) the answers of the following eight parts neatly. Do rough work at the back of your answer sheet. **[4+4+4+4+4+4+4+8 = 36M]**

a) If A = $(-131)_{10}$ and B = $(231)_{10}$, what is the value of C if (A+B) = $(C)_{16}$? Assume 16-bit 2's complement number representation here.

b) If A = $(-131)_{10}$ and B = $(231)_{10}$, what is the maximum positive value of C in base-10 if the following expression does not result in overflow: **((A AND B) + C)**. Assume 16-bit, 2's complement number representation here.

c) Let A = 10111101 be an unsigned binary number. Convert it into base-6 number representation.

d) Convert the following 32-bit IEEE 754 number to its equivalent decimal number:

1 10001110 01010000000000000000000

e) While converting any decimal number to IEEE 754 floating point number representation, why we get biased exponent by adding 127 to actual exponent?

f) Let there be an 8-digit positive integer whose individual digits are stored in an array of size 8 with least significant digit stored at index 7 and most significant digit at index 0. Complete the program (shown in **FIGURE-1**) which stores the reverse of this 8-digit integer in variable **sum**. For example, positive integer 32406321 is stored in an array as [3,2,4,0,6,3,2,1]. After program execution, variable **sum** stores the number 12360423.

```
#define nod 8
int main()
{
   int arr[nod], N;
/*Assume arr is populated with individual
digits of 8-digit integer N as explained in Q1f*/
   int i, sum = 0;
   for (i=nod ; i>0 ; i--)
/* Write a single statement here to get
the desired result in sum. Just write this
statement in your answer sheet.*/
   printf ("%d", sum);
}
```
FIGURE-1

g) What is the output of two programs shown in **FIGURE-2** and **FIGURE-3**?

h) Consider an n-digit number $d_1d_2……d_n$ which is to be stored in a 1D array **Lin** of size n. From this 1D array it is required to populate a 2D array **Arr** of size **nxn** as follows: The 0^{th} row of **Arr** should contain $d_1d_2……d_n$. The 1^{st} row should contain $d_2d_3…d_n\ d_1$. The 2^{nd} row should contain $d_3d_4……d_nd_1d_2$ and so on. The $(n-1)^{th}$ row of **Arr** should contain $d_nd_1d_2……d_{n-1}$. In the program (shown in **FIGURE-4**), what should be substituted in place of indexes N1 to N6 to get the desired result?

```
int main() {
   int n, i, j, k;
   printf("\nEnter no. of digits");
   scanf("%d", &n);
   int Lin[n], Arr[n][n];
   for(i=0;i<n;i++) //Input each digit
      scanf ("%d",&Lin[i]);
   for(i=0; i<n; i++)
   {
      for(j = 0;j<n-i;j++)
         Arr[N1][N2] = Lin[N3];
      for (k=0;k<i;k++)
         Arr[N4][N5] = Lin[N6];
   }
   return 0;
}
```
FIGURE-4

```
int main()
{
   int i=0;
   for (;i<20;i++)
   {
      switch (i)
      {
         case 0: i+=5;
         case 1: i+=2;
         case 5: i+=5;
         default: i+=4;
      }
      printf ("%d\n",i);
   }
   return 0;
}
```
FIGURE-2

```
int fun ()
{
   int i=0;
   for (;i<20;i++)
   {
      switch (i)
      {
         case 0: i+=5;
         case 1: i+=2;
         case 5: i+=5;
         default: i+=4;
      }
      return i;
   }
}
int main()
{
   int j = fun();
   printf ("%d\n", j);
   return 0;
}
```
FIGURE-3

Q2. It is required to write a program for an ATM machine which dispenses currency notes in denominations of Rs 2000, Rs 500, and Rs 100. The user is asked to enter the desired amount (a multiple of Rs 100) and the ATM machine dispenses this amount using minimum number of notes. Complete the following program for it: [**10M**]

```
void main()
{
   int amount, no_of_2000, no_of_500, no_of_100;
   printf ("Enter amount");
   scanf ("%d",&amount);

   /*Implement a function, named get_denominations(), and call it here. This function computes minimum
number of notes required of each denomination for desired amount and stores them in variables no_of_2000,
no_of_500, and no_of_100 in main. Do not use any global variables.*/

   printf ("Number of Rs 2000 notes = %d\n",no_of_2000);
   printf ("Number of Rs 500 notes = %d\n",no_of_500);
   printf ("Number of Rs 100 notes = %d\n",no_of_100);
}
```

Q3. See the following three programs (P1, P2, and P3) and their corresponding outputs carefully. There is a feature of C-language because of which all three programs are giving such surprising outputs. Deduce and explain this feature with respect to any one program (P1, P2, or P3). Assume that **int** data type is stored using 2's complement number representation. [**14M**]

```
/*Program P1. Its output is
GOTIT !   */
int main()
{
   unsigned int num1 = 10;
   int num2 = -100;
   if ((num1+num2) > 10)
      printf ("GOTIT !");
   else
      printf ("MAGIC");
}
```

```
/* Program P2. Its output is
GOTIT BITS    */
int main()  {
   unsigned int a = -1;
   int b = 1;
   if (a == -1)  printf ("GOTIT ");
   else  printf ("MAGIC");

   if (a > b)  printf ("BITS");
   else  printf ("PILANI");
}
```

```
/* Program P3. This program is intended to print
the elements of array arr[]. But, surprisingly, it
will not print anything. */

   int arr[] = {1,2,3,4,5}; //globally defined array
   #define NUM (sizeof(arr) / sizeof(arr[0]))
    //Assume int require 4 bytes of memory.
    //sizeof(arr) returns 20, sizeof arr[0] returns 4.
    //Therefore, NUM contains value 5
   void main() {
      int i;
      for(i=-1;i <= (NUM-2);i++)
         printf("%d\n",arr[i+1]);
   }
```

Q4. Consider an incomplete program shown in **FIGURE-5** that reads N integers to populate the elements of an array **data[N].** Further, this program finds the largest sum (to be stored in variable **max**) of consecutive array elements and a corresponding range of index values (to be stored in variables **startIndex** and **endIndex**). Finally, this program prints the value of **max** and the array elements which produce the largest sum. Complete the program ***using only for loops***.

For example, for the following array of 7 integers the largest sum of consecutive array elements is 8 which we get by adding elements from **data[1]** to **data[5]** (i.e. 2, 3,-2, 0, 5).

-10	2	3	-2	0	5	-15

[Note: You can declare additional variables if needed**]**

[18M]

```
#define N 7
int main() {
   int data[N], index, startIndex, endIndex, max;
   printf("Enter array elements\n");
   for(index=0;index<N;index++)
         scanf("%d",&data[index]);
   max=data[0];
   startIndex = endIndex = 0;

      //Write your code here to complete the program

   printf("Maximum sum = %d\n",max);
   for(index=startIndex; index<=endIndex; index++)
     printf("%d",data[index]);
   return 0;
}
```

FIGURE-5