Indian Institute of Technology Kharagpur
Department of Computer Science and Engineering

CS10003 Programming and Data Structures

Spring Semester, 2021-2022 **Sections 5 & 6**

Long Test 1 – Part B[Marks = 50] **18-May-2022, 10:15 to 11:25**

INSTRUCTIONS

- You have 1 hour for writing and 10 minutes for submission.
- Write your answers on paper. Answers must be handwritten. Typed or written answers using an electronic device are not allowed.
- Write your name and roll number on each page. Write page numbers for each page.
- Scan all pages and collate. Create a single PDF file (of size ≤ 10 MB). You could also take pictures of different pages, combine them to make a single pdf file.
- The name of the file for this part should be <RollNumber>_Long1B. Ensure your roll number in the filename has only digits and uppercase characters.
- Upload your file. Make sure you click on the 'Turn in' button to submit your file.
- The said deadlines are strict after which no submissions will be allowed.
- We will not accept submission by any other means.

1. Write a C program to read a positive integer n, $n < 20$, read n positive integers and store them in an array. Then reorder the array by having all the odd numbers followed by all the even numbers. Print the reordered array. The rearranged even and odd numbers may be in any order among themselves. For example, if $n = 6$ and the numbers given are 43, 78, 56, 13, 9, 8 then one possible output is 43, 9, 13, 56, 78, 8. Do not use any library functions other than standard input and output. You can use only ONE array of size at most 20. There should be only one main function and no other functions defined by you. $\boxed{10}$

Solution:

```
#include<stdio.h>
main()
{
    int A[20], n, i, j, k, temp;
    scanf("%d", &n);
    printf("n = %d \n", n);
    for (i=0; i<n; i++) scanf("%d", &A[i]);
    printf("Original A = ");
    for (i=0; i<n; i++) printf("%d,  ", A[i]);
    printf("\n");
    i = 0;
    j = n-1;
    while(i<=j)
    {
        while (A[i]%2 == 1) i++;
        while (A[j]%2 == 0) j--;
        printf("i = %d, A[i] = %d, j = %d, A[j] = %d \n", i, A[i], j, A[j]);
        if (i < j)
        {
            temp = A[i];
            A[i] = A[j];
            A[j] = temp;
        }
    }
    printf("Rearranged A = ");
    for (i=0; i<n; i++) printf("%d,  ", A[i]);
    printf("\n");
}
```

2. In the normal scale of Indian music, we have the notes, Sa, Re, Ga, Ma, Pa, Dh, Ni. Note that each note is represented by two characters. A sequence of notes is a palindrome note, if it reads the same when we read the sequence from right-to-left and when we read the sequence from left-to-right. For example, the sequence Ga Pa Dh Ni Dh Pa Ga is a palindrome note sequence. Also Re Re Ma Ma Re Re is a palindrome note sequence. Write a program that reads an integer n followed by a sequence of n notes and prints whether the note sequence is a palindrome note sequence. Assume $n \leq 50$. Your program must have only one *main*() function and no other functions. $\boxed{10}$

Solution:

```
#include <stdio.h>
#include <string.h>

int main(){
  char note[3];
  char seq[51];
  int n, i, j, len;
  scanf("%d", &n);
  printf("Enter note sequence with each note in a separate line:\n");
  j=0;
  for(i=1; i<=n; i++){
    scanf("%s", note);
    seq[j++] = note[0];
  }
  seq[j] = '\0';
  len = j;
  for(i=0; i<len/2; i++){
    if(seq[i] != seq[len-1-i]){
      printf("Not a palindromic note sequence.\n");
      return 0;
    }
  }
  printf("The given sequence is palindromic.\n");
  return 0;
}
```

3. Write a function *print_bits* that takes as input a positive integer n and prints the binary representation of n with a single space separating every two consecutive bits. Do not use arrays/strings or any global variables. Within the *print_bits* function, you can declare and use at most 4 integer variables (other than n). In the *main*() function, read an integer n and call *print_bits* on input n. For example, if the input is 44, the output should be 1 0 1 1 0 0. Do not use any library functions other standard input/output. $\boxed{10}$

Solution:

```
#include <stdio.h>

void print_bits(int n){
  int nzero = 0;
  int m=0;
  while(n%2 == 0){
    n = n/2;
    ++nzero;
  }
  while(n!=0){
    m *= 2;
    m = m+(n%2);
    n /= 2;
  }
  while(m!=0){
    printf("%d ",m%2);
    m = m/2;
  }
  while(nzero != 0){
    printf("0 ");
    nzero--;
  }
  printf("\n");
}

int main(){
  int n;
  printf("n = ");
  scanf("%d", &n);
  printf("Binary representation of %d: ",n);
  print_bits(n);
  return 0;
}
```

4. Following is a function `void triangle(int n)`, which is written partially. The function would print for an input, say rows = 6 (reading from the keyboard) the following.

```
           1
         1   1
       1   2   1
     1   3   3    1
   1  4    6   4   1
 1  5   10   10  5   1
```

```
void triangle(int n) {
   int print = 1, gap, i, j;
   for (i = 0; _________; i++) {                  (A)
      for (gap = 1;________________; gap++)       (B)
         printf("  ");
      for (j = 0; ___________; j++) {             (C)
         if (________________)                    (D)
            print = 1;
         else
            print = ___________________;          (E)
         printf("%4d", print);
      }
```

```
        printf("\n");
    }
    return 0;
}
```

10

Solution:

```
    (A) i < n
    (B) gap <= n - i
    (C) j<= i
    (D) j == 0 || i == 0
    (E) print*(i - j + 1) / j
```

5. Write a recursive function that on input two integers n, r computes the following function.

$$C(n,r) = \begin{cases} 0 & \text{if } n < r \\ 1 & \text{if } r = 0 \text{ and } n \geq r \\ C(n-1,r) + C(n-1,r-1) & \text{if } r > 0 \end{cases}$$

Read non-negative integers n and r as inputs in the $main()$ function ensuring that $n \geq r \geq 0$. Call the above function and print n, r and $C(n,r)$. 10

Solution:

```
# include <stdio.h>

int C (int n, int r)  {
  if (r == 0)
    return(1);
  else if (n==r)
    return(1);
  else if (n > r) {
    return(C(n-1, r)+ C(n-1, r-1));
  }
}

int main () {
  int N,R;
  printf ("\nEnter N, R : ");
  scanf ("%d %d", &N,&R);
  if(R < 0 || N < R){
    printf("Invalid input\n");
    return 0;
  }
  printf("Answer = %d\n",C(N,R));
}
```