

CS10003 Programming and Data Structures

Spring Semester, 2021-2022

Sections 1 & 2

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. Consider the following program fragment, where $i, j, count$ are int type variables, and a, b and c are 50 element character arrays. What will be printed if we enter the strings “Programming”, “Problems” and “Data” from the keyboard (in that order)?

```
scanf("%s%s%s", a, b, c);
i = 0; count = 0;
while (a[i] != '\0') {
    j = 0;
    while (b[j] != '\0') {
        if (a[i] != b[j]) {
            j++;
            continue;
        } else {
            count++;
            break;
        }
    }
    i++;
}
printf ("%d", count);
```

Solution: 6

2. We know that in a sorted array A of n distinct integers, $A[i]$ has rank i , that is, $A[i]$ is larger than exactly i elements in the array A , as in the example for $n = 8 = 2^3$, $A = \{2, 4, 47, 56, 99, 100, 120\}$. If A is not sorted as for the case $n = 4$ and $A = \{2, 1, 5, 7\}$, $A[0]$ has rank 1 and not 0.

(a) Write a C program to sort an unsorted given array A of $n = 2^k$ distinct integers by computing the rank of each element in A , where k is any positive integer. So, you need to read n , and the entire unsorted array A of n distinct integers, sort the array A , and print the sorted array. You must also print the rank of all the n elements of A .

The shuffle operation on an array A is as follows, where we assume that $n = 2^k$. For all even i , $0 \leq i \leq n - 2$, we must place these numbers in the same order in the first half of the array A , and for all odd i , $1 \leq i \leq n - 1$, we must place $A[i]$ in the same order in the second half of the array. So, for $A = \{1, 3, 4, 56, 77, 111, 222, 8888\}$, we must transform A so that A becomes $\{1, 4, 77, 222, 3, 56, 111, 888\}$.

(b) Your program must further shuffle the sorted array of part (1), k times where we assume that $n = 2^k$, for any positive integer k . After each shuffle, you must print the array. Shuffling the above array $A = \{1, 4, 77, 222, 3, 56, 111, 888\}$ gives $A = \{1, 77, 3, 111, 4, 222, 56, 888\}$.

10

Solution:

```
#include<stdio.h>

int main (){

    int A[100],B[100],n,i,j,rank[100];

    printf("Enter an integer\n");
    scanf("%d",&n);
    printf("Enter %d distinct integers \n",n);

    for (i=0;i<n;i++)
        scanf("%d",&A[i]);

    for (i=0;i<n;i++) {
        rank[i]=0;
        for (j=0;j<n;j++) {
            if (A[j]<A[i])
                rank[i]++;
        }
        B[rank[i]]=A[i];
    }

    for (i=0;i<n;i++)
        A[i]=B[i];
    printf("The sorted array is\n");
    for (i=0;i<n;i++)
        printf("%d ",A[i]);
    printf("\n");

    printf("The rank array is\n");
    for (i=0;i<n;i++)
        printf("%d ",rank[i]);
    printf("\n");
```

```

int nn; int c = 1;
for (nn=n;nn!=1;nn=nn/2) {
    for (i=0;i<n;i++) {
        if (i%2==0)
            B[i/2]=A[i];
        else
            B[n/2+i/2]=A[i];
    }
    for (i=0;i<n;i++)
        A[i]=B[i];
    printf("The array after shuffle %d is\n", c);
    for (i=0;i<n;i++)
        printf("%d ",A[i]);
    printf("\n");
    c++;
}
}

```

3. Write a C function that has three positive integer arguments n, m, k and returns the third largest integer p , $p \leq k$, such that p divides both n and m . It returns 0 if there is no such third largest $p > 0$. For example, $n = 200$, $m = 100$, $k = 80$, result is 20. Again, if $n = 35$, $m = 25$, $k = 18$, then the result is 0. Do not use arrays or any library functions other than standard input and output. Write the function and the main program. In the main program, read the inputs, call the function and print the result.

10

Solution:

```

#include<stdio.h>
int third(int n, int m, int k)
{
    int count, i;
    count = 0;
    for(i = k; i > 0; i--)
    {
        if ((n%i == 0) && (m%i == 0)) count++;
        if (count == 3) break;
    }
    if (count == 3) return (i);
    else return (0);
}

main()
{
    int n, m, k;
    scanf("%d%d%d", &n, &m, &k);
    printf("n = %d, m = %d, k = %d \n", n, m, k);
    printf("Result = %d \n", third(n,m,k));
}

```

4. Write a function which takes as input two integer arrays, $a[]$ and $b[]$, of size n and prints only those numbers which appear in both the arrays. Assume that for each array, the elements are provided in ascending order. Each array stores n elements, with no duplication of the elements. Also, assume that $n \leq 100$.

10

Solution:

5. Write a recursive function that takes as input an integer n and returns the sum of all its digits that are ≤ 5 . For example, on input 934072856, the output should be 14. In the `main()` function, read an integer x and print the whatever the function returns on input x .

10

Solution:

```
#include <stdio.h>

int sumDigits5(int n){
    int sum = 0, d;
    if(n == 0)
        return 0;
    if(n%10 <= 5)
        sum = sumDigits5(n/10) + (n%10);
    else
        sum = sumDigits5(n/10);
    return sum;
}

int main(){
    int x;
    printf("x = ");
    scanf("%d", &x);
    printf("%d\n",sumDigits5(x));
    return 0;
}
```