

INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
END SEMESTER EXAMINATION, AUTUMN 2012-13

Programming & Data Structures (CS 11002)

Full marks: 100

Time: 3 hours

1. Answer all questions in the space provided in this question paper itself. No extra sheet will be provided.
2. As mentioned in class, questions have been shuffled across different question papers.
3. No further clarifications to any of the questions will be provided.

Name	Roll No.	Section	Marks obtained

1. What would be the output produced when the following code fragment is executed? [2]

```
int arr1[]={0,1,1,0,0,1}; int *pt1=arr1;
while(*pt1++);
printf("%d\n", 499747*(pt1-arr1));
```

2. Declare an array of three strings and initialize it (in the declaration statement itself) to contain the following three strings:
cat, dog, giraffe. [2]

3. What will be output of the following code? [2]

```
#include<stdio.h>
int main(){
    int i=1;
    for(i=0;i=-1;i=1) {
        printf("%d11111\n",i);
        if(i!=1) break;
    }
    return 0;
}
```

4. What will be displayed after the following program fragment executes? [2]

```
char saying[] = "Too many cooks spoil the broth.";
char *p1, *p2;
p1 = saying;
p2 = saying + 8;
*p2 = '\\0';
printf("%s\\n", saying);
```

5. A number is said to be *perfect* if it is the sum of all its possible factors (except itself!). For example, 6 has factors 1, 2, 3, and $1 + 2 + 3 = 6$, and hence 6 is perfect. Similarly, $28 = 1 + 2 + 4 + 7 + 14$ is also perfect. Fill up the missing parts so that the following function `checkPerfect` returns 1 if the number given as argument is perfect, and returns 0 if it is not. [2]

```
int checkPerfect(int n){
    int sum = 0;
    for(int i=1; i<n; i++)

        if ( _____ == 0)

            sum += i;

    return (_____);
}
```

6. 18,000 character strings denoting the names of 18,000 persons are arranged in lexicographic ordering in an array of strings. Binary search is applied on this array to check whether a given name is present in the array. In the binary search procedure, the function `strcmp` is used to compare two strings. Determine the maximum number of times the `strcmp` function would get invoked. [3]

7. What should be the output of the following program? [3]

```
#include<stdio.h>
int f(int x){
    if (x<5) return x;
    printf("%d: ",248+x);
    return (f(--x));}
int main(){
    printf("%d\n",11107+f(7)); return 0;}
```

8. Consider the following function myStrcat. Its parameters are pointers to two strings x and y (which have large enough allocated space), and the function is supposed to concatenate the string y after the string x. Fill up the missing parts, so that it behaves as required. [3]

```
void myStrcat(char *x, char *y){
    char *i, *j;

    for (_____ ; _____ ; _____) //2 marks

        ; // empty body

    for (j = y; *j; ++j, ++i)

        _____; //assignment expression - 1 mark
}
```

9. In a certain computer, the size of an integer is 4 bytes and each memory address is of 8 bytes. What will be displayed when the following program segment executes? [3]

```
int main(){
    int a[] = {12, 13, 14, 15, 16};
    printf("sa %d, sa* %d, sa[] %d\n", sizeof(a), sizeof(*a), sizeof(a[0]));
    return 0;
}
```

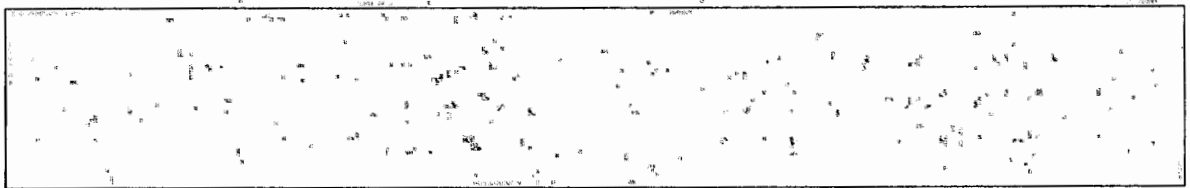
10. What will be the output when the following program segment is executed? [3]

```
int main() {
    char c;
    for (c='a'; c<'g'; ++c) {
```

```

switch (c) {
    case 'a': c += 2;
    case 'c': c += 1;
    case 'g': ++c; printf ("%c\n", c--);
    default: ++c;
}
printf ("***%c\n", c);
} //end for
return 0;
} //end main

```

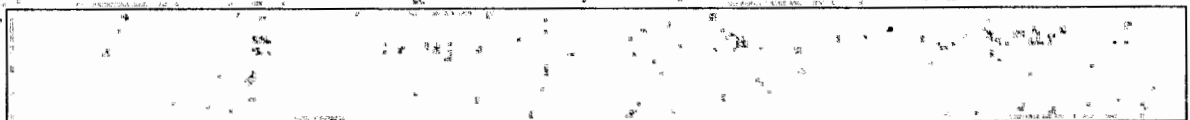


11. The following function `removeLast` is supposed to remove the last non-null character of the string `str` that it gets as its parameter. However, the function has been observed to behave erratically on some inputs. Underline the statement that has a bug, and write the corrected code in the box provided below. [3]

```

void removeLast(char *str){
    int i=0;
    while (str[i] != '\0')
        i = i + 1;
    str[--i] = '\0';
}

```

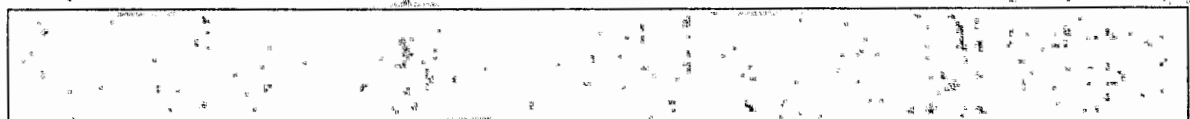


12. What will be the output produced when the following code segment executes? [3]

```

float A[6] = {1.0, 2.0, 1.0, 0.5, 3.0, 2.0};
float *t = &(A[0]);
float **it = &t, *w = &(A[1]);
while (w < &(A[6])) {
    if (*w < *t) *it = w;
    w += 1;
}
printf("%.1f%d\n", *t, 248698/2);

```



13. In the following code segment, the integer array A has been initialized to contain the values 1, 2, 3, 4, 5. Fill in the dotted lines of the code so that each element of the array A is moved to the immediate next position in the array. The loop should discard the last value in the array (i.e., 5), and the array should contain the values 1, 1, 2, 3, 4 after the execution of the loop. [3]

```
int k, A[5]={1,2,3,4,5};

for (k=4; _____; _____) {

    A[k] = _____;

}
```

14. Write a function HasZeroDiagonal that, given a 5-by-5 integer array, returns 1 if all the elements on the main diagonal of the array are zero, and returns 0 otherwise. [3]

```
int HasZeroDiagonal (int a[][]) {
    int k, result = 1;

    for (k=0; _____; k++) //1 mark

        result = result && (a[_____][_____] == 0); //2 marks

    return result;
}
```

15. Write the number of exchanges or swaps that will take place during Bubble Sort on the following sequence of numbers: 4, 8, 6, 2, 5, 9, 7. [4]

16. What will be displayed after the following code segment executes? [4]

```
int A[]={2,6,5,1,3}, *p=A;
printf("2 2 1 2 2 1 ", );
printf("%d ", *p);
printf("%d ", *p+(*(p+4)-3));
printf("%d ", *(p+2));
printf("%d\n", *p+2);
```

17. What will be displayed after the following code segment executes? [4]

```
int a[]={1,2,3,4,5,6}, *iP=a+4;
printf("2 2 1 2 2 1 ", );
printf("%d ", iP[1]);
printf("%d\n", iP[a[iP[-3]]-3]);
```

18. Consider the following declaration of `studArr`, an array of 600 structures. Assume that all the structures in the array have been populated by reading the corresponding data from the keyboard. [4]

```
struct student{
    char Name[11]; char Address[21]; int Roll_number; float CGPA;
} studArr[600];
```

Consider the following function named `elimDuplicate` that takes no argument, and checks duplicate entries by matching the names in `studArr`. All those duplicate entries are eliminated by moving forward the elements of `studArr` after a deleted entry. Fill up the missing code for making the function `elimDuplicate` behave as required. Note that by the word “duplicate”, we mean that a record can occur at most twice in the array.

(Hint: You may use any string library function, if you wish.)

```
void elimDuplicate(){
    int i, j, k, last=600;
    for(i=0;i<last;i++)
        for(j=i+1;j<last;j++)

            if ( _____ == 0){
                                                                    /* if match */

                /* shift the rest one place forward */

                for ( _____)

                    _____;

                last--; /* one record less */
            }
}
```

19. What will be output of following C code? [4]

```
#include<stdio.h>
int r();
int main(){
```

```

    for(r();r();r())
        printf("%d449\n",r());
    return 0;
}
int r(){
    int static num=7;
    return num--;}

```

20. The following function reverseString takes a string as argument and is expected to return the reversed string. However, it does not work as intended. Using a debugger, the error was localized to the statement of the for loop. And one more sentence needs to be added after the for loop to make it work as intended. Write the corrected statement in the box provided below, and add the next sentence in the dotted line. [4]

```

char *reverseString(char *src) {
    int j;
    char *dest;
    dest = (char *) malloc(strlen(src) + 1);
    for (j = 0; j < strlen(src); j++)
        dest[strlen(src)-j] = src[j]; //correct the error

```

```

----- //write the missing sentence
return dest;
}

```

21. Fill in the blanks in the program below so that it displays the contents of the file whose name is provided by the user. [5]

```

#include <stdio.h>
#include <stdlib.h>
int main() {
    char c, fileName[32]; FILE *inFile;
    printf ("File name? "); scanf ("%s", fileName);

    inFile = fopen (_____, "r"); // 1 mark

    if (_____ == NULL) { // 1 mark

```

```

    fprintf (stderr, "File doesn't exist!\n"); exit(1);}

while ((c=getc(inFile)) != EOF) // 1 mark

    fprintf (stdout, " "); // 2 marks

return 0;
}

```

22. Fill up the gaps to define a struct to represent dates. The struct members are: (i) a string representing the month, (ii) an integer representing the date in the month, and (iii) an integer representing the year. Then define the type Date to be synonymous with the struct definition, and define an array dateList of five dates. [5]

```

struct date {
    _____ month[10]; /* September is the longest month name. */
    _____ dateInMonth;
    int year;
};

typedef _____ date;

Date _____;

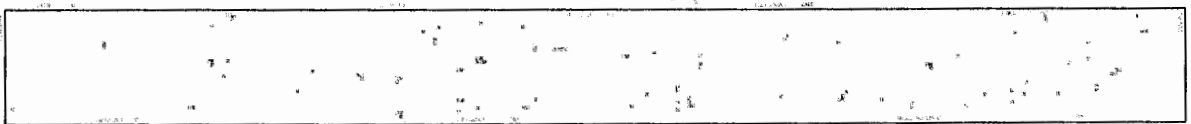
```

23. What would be the output produced by the following segment of C code? [5]

```

int x, j, p[10], *q=p;
*p = 4;
for (j=0; j<10; j++){
    x= *q++; *q= j+x;}
q=p;
for (j=0; j<10; j++)
    printf("%d-",*q++);

```



24. Given the following declarations:

```

struct tnode {
    char word[10];

```



```

    struct tnode *left, *right;
};
typedef struct tnode TREENODE, *TREETPTR;
TREETPTR tree;
TREENODE node;

```

Now from the following six expressions, strike out those which are not legal expressions. [4]

```

node->word[2]
node.word[2]
tree->left->word
*(tree->left)
*tree->left
tree->*left

```

25. What will be the output of the following program? [5]

```

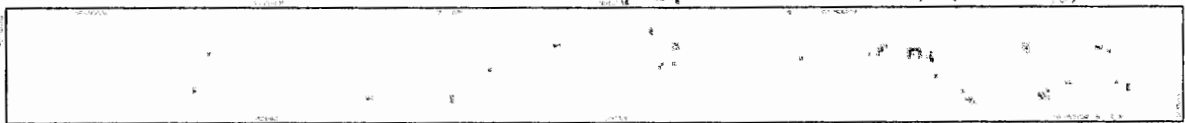
#include <stdio.h>
int main(){
    int i, j, k;
    for (i=1; i<10; i++) {
        printf("\n %d: ", i);
        for (j=1; j<10; j++) {
            if (i%3 == 0) break;
            if (i > j) continue;
            k=i*10+j;
            printf(" %d", k);
        }
    }
}

```

26. What will be the output produced by the following program?

[5]

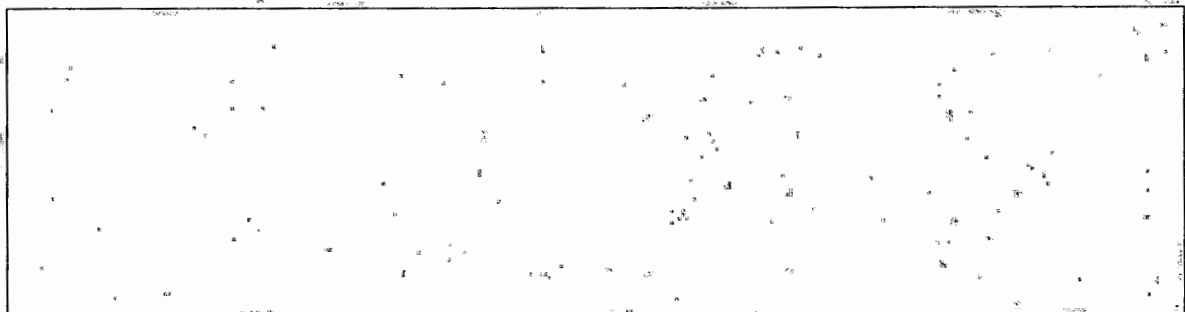
```
#include<stdio.h>
int main(){
    int i, n=10, k=0, A[10]={0,1,2,3,4,5,6,7,8,9};
    int lim = n/2;
    for(i=0;i<lim;i++){
        k = k + A[i] + A[n-i-1];
        if(lim < (n-lim))
            k = k + A[i];
        printf("99%d12.71\n", k);
    }
    return 0;
}
```



27. What is the output of the following code fragment?

[5]

```
int array[4][4], index1, index2;
for(index1=0;index1<4;index1++){
    for(index2=0;index2<4;index2++){
        array[index1][index2] = index1 + index2;
    }
}
for(index1=0;index1<4;index1++){
    for(index2=0;index2<4;index2++){
        printf("%d ", array[index1][index2]);
        printf("\n");
    }
}
```



28. For the following program, write the output.

[5]

```
#include <stdio.h>
int main(){
    float sum = 0.0, j = 1.0, i = 2.0;
    while (i/j > 0.0625){
```

```
j = j + j;  
sum = sum + i/j;  
printf("%f - ", sum);  
printf("\n", sum);  
return 0;  
}
```

