

# Birla Institute of Technology & Science, Pilani Hyderabad Campus

Second Semester 2020-2021

## Computer Programming [CS F111]

### Lab 9 (Arrays)

**Q1.** Write a program to find out largest number from a list of numbers stored in an Array.

```
1  #include <stdio.h>
2
3  int main( )
4  {
5      float largest(float a[ ], int n);
6      float value[4] = {2.8,-4.75,3.2,3.67};
7      printf("%f\n", largest(value,4));
8  }
9  float largest(float a[], int n)
10 {
11     int i;
12     float max;
13     max = a[0];
14     for(i = 1; i < n; i++)
15         if(max < a[i])
16             max = a[i];
17     return(max);
18 }
```

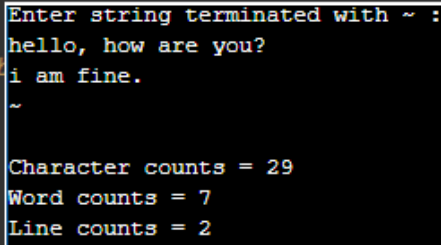
3.670000

**Q2.** Write a program to input a set of characters, words, lines into an Array terminated by ~. Find out how many words, lines and characters are there in the input.

```

1  #include<stdio.h>
2  int main()
3  {
4      char str[200];
5      int line, word, ch;
6
7      line = word = ch = 0;
8      printf("Enter string terminated with ~ :\n");
9      scanf("%[^~]", str);
10     // check every character
11     for(int i=0; str[i]!='\0'; i++)
12     {
13         // if it is new line then one line and one word completed
14         if(str[i]=='\n')
15         {
16             line++;
17             word++;
18         }
19         // else it is a character
20         else
21         {
22             // if character is space or tab then one word is also completed
23             if(str[i]==' ' || str[i]=='\t')
24             {
25                 word++;
26                 ch++;
27             }
28             // it was not '\n', space or tab
29             // it is a normal character
30             else {
31                 ch++;
32             }
33         }
34     }
35     printf("\nCharacter counts = %d\n", ch);
36     printf("Word counts = %d\n", word);
37     printf("Line counts = %d\n", line);
38     return 0;
39 }

```



```

Enter string terminated with ~ :
hello, how are you?
i am fine.
~

Character counts = 29
Word counts = 7
Line counts = 2

```

**Q.3** Write a C program to implement Booth's Multiplication Algorithm for multiplication of 2 signed numbers

**Explanation:**

This is a C Program to multiply two signed numbers using booth's algorithm. Booth's multiplication algorithm is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation. Booth used desk calculators that were faster at shifting than adding and created the algorithm to increase their speed. Booth's algorithm is of interest in the study of computer architecture.

**Algorithm:**

Put multiplicand in BR and multiplier in QR and then the algorithm works as per the following conditions :

1. If  $Q_n$  and  $Q_{n+1}$  are same i.e. 00 or 11 perform arithmetic shift by 1 bit.
2. If  $Q_n Q_{n+1} = 10$  do  $A = A + BR$  and perform arithmetic shift by 1 bit.
3. If  $Q_n Q_{n+1} = 01$  do  $A = A - BR$  and perform arithmetic shift by 1 bit.

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int a = 0, b = 0, c = 0, a1 = 0, b1 = 0, com[5] = { 1, 0, 0, 0, 0 };
5  int anum[5] = {0}, anumcp[5] = {0}, bnum[5] = {0};
6  int acomp[5] = {0}, bcomp[5] = {0}, pro[5] = {0}, res[5] = {0};
7
8  void binary(){
9      a1 = fabs(a);
10     b1 = fabs(b);
11     int r, r2, i, temp;
12     for (i = 0; i < 5; i++){
13         r = a1 % 2;
14         a1 = a1 / 2;
15         r2 = b1 % 2;
16         b1 = b1 / 2;
17         anum[i] = r;
18         anumcp[i] = r;
19         bnum[i] = r2;
20         if(r2 == 0){
21             bcomp[i] = 1;
22         }
23         if(r == 0){
24             acomp[i] = 1;
25         }
26     }
27     //part for two's complementing
28     c = 0;
29     for (i = 0; i < 5; i++){
30         res[i] = com[i] + bcomp[i] + c;
31         if(res[i] >= 2){
32             c = 1;
33         }
34         else
35             c = 0;
36         res[i] = res[i] % 2;
37     }
38     for (i = 4; i >= 0; i--){

```

```

39         bcomp[i] = res[i];
40     }
41     //in case of negative inputs
42     if (a < 0){
43         c = 0;
44         for (i = 4; i >= 0; i--){
45             res[i] = 0;
46         }
47         for (i = 0; i < 5; i++){
48             res[i] = com[i] + acomp[i] + c;
49             if (res[i] >= 2){
50                 c = 1;
51             }
52             else
53                 c = 0;
54             res[i] = res[i] % 2;
55         }
56         for (i = 4; i >= 0; i--){
57             anum[i] = res[i];
58             anumcp[i] = res[i];
59         }
60     }
61 }
62 if(b < 0){
63     for (i = 0; i < 5; i++){
64         temp = bnum[i];
65         bnum[i] = bcomp[i];
66         bcomp[i] = temp;
67     }
68 }
69 }
70 void add(int num[]){
71     int i;
72     c = 0;
73     for (i = 0; i < 5; i++){
74         res[i] = pro[i] + num[i] + c;
75         if (res[i] >= 2){
76             c = 1;

```

```

77         }
78     }
79     else{
80         c = 0;
81     }
82     res[i] = res[i]%2;
83 }
84 for (i = 4; i >= 0; i--){
85     pro[i] = res[i];
86     printf("%d",pro[i]);
87 }
88 printf(":");
89 for (i = 4; i >= 0; i--){
90     printf("%d", anumcp[i]);
91 }
92 }
93 void arshift(){//for arithmetic shift right
94     int temp = pro[4], temp2 = pro[0], i;
95     for (i = 1; i < 5 ; i++){//shift the MSB of product
96         pro[i-1] = pro[i];
97     }
98     pro[4] = temp;
99     for (i = 1; i < 5 ; i++){//shift the LSB of product
100         anumcp[i-1] = anumcp[i];
101     }
102     anumcp[4] = temp2;
103     printf("\nAR-SHIFT: ");//display together
104     for (i = 4; i >= 0; i--){
105         printf("%d",pro[i]);
106     }
107     printf(":");
108     for(i = 4; i >= 0; i--){
109         printf("%d", anumcp[i]);
110     }
111 }
112 void main(){
113     int i, q = 0;
114     printf("\t\tBOOTH'S MULTIPLICATION ALGORITHM");

```

```

115     printf("\nEnter two numbers to multiply: ");
116     printf("\nBoth must be less than 16");
117     //simulating for two numbers each below 16
118     do{
119         printf("\nEnter A: ");
120         scanf("%d",&a);
121         printf("Enter B: ");
122         scanf("%d", &b);
123     }while(a >=16 || b >=16);
124
125     printf("\nExpected product = %d", a * b);
126     binary();
127     printf("\n\nBinary Equivalents are: ");
128     printf("\nA = ");
129     for (i = 4; i >= 0; i--){
130         printf("%d", anum[i]);
131     }
132     printf("\nB = ");
133     for (i = 4; i >= 0; i--){
134         printf("%d", bnum[i]);
135     }
136     printf("\nB' + 1 = ");
137     for (i = 4; i >= 0; i--){
138         printf("%d", bcomp[i]);
139     }
140     printf("\n\n");
141     for (i = 0; i < 5; i++){
142         if (anum[i] == q){//just shift for 00 or 11
143             printf("\n-->");
144             arshift();
145             q = anum[i];
146         }
147         else if(anum[i] == 1 && q == 0){//subtract and shift for 10
148             printf("\n-->");
149             printf("\nSUB B: ");
150             add(bcomp);//add two's complement to implement subtraction
151             arshift();
152             q = anum[i];
153         }
154         else{//add ans shift for 01
155             printf("\n-->");
156             printf("\nADD B: ");
157             add(bnum);
158             arshift();
159             q = anum[i];
160         }
161     }
162
163     printf("\nProduct is = ");
164     for (i = 4; i >= 0; i--){
165         printf("%d", pro[i]);
166     }
167     for (i = 4; i >= 0; i--){
168         printf("%d", anumcp[i]);
169     }
170 }

```

**Output:**

```
BOOTH'S MULTIPLICATION ALGORITHM
Enter two numbers to multiply:
Both must be less than 16
Enter A: 12
Enter B: 12

Expected product = 144

Binary Equivalents are:
A = 01100
B = 01100
B'+ 1 = 10100

-->
AR-SHIFT: 00000:00110
-->
AR-SHIFT: 00000:00011
-->
SUB B: 10100:00011
AR-SHIFT: 11010:00001
-->
AR-SHIFT: 11101:00000
-->
ADD B: 01001:00000
AR-SHIFT: 00100:10000
Product is = 0010010000

...Program finished with exit code 0
Press ENTER to exit console.
```

**Q.4** Assume that Admission division at BITS, Pilani wants you to write a C program to admit students into its 3 engineering streams (Civil, EEE and CSE) based on the following logic for the year 2021. Assume that there are 2 seats in each of these branches. Your program should receive as input student application number (an integer), his scores as per the below requirement and allot him/ her a branch according to his/ her preferences and as per the admission rules given below:

Admission logic:

1. A student should have a minimum aggregate of 75% marks in Phy, Chem and Maths; and 60% in each of the PCM subjects at 12th standard. He/ she should have passed 12th standard either in 2021 or in 2020.
2. He/ she should have scored a minimum of 240 marks in the current year's (2021) BITSAT.
3. Seats are allotted to students on the basis of their BITSAT scores while PCM scores in 12th are considered as mere eligibility, i.e. if a student with a valid BITSAT score wants first a seat in a particular stream, then he/ she is allotted the seat provided his MSN (merit serial number) is higher than others.

4. Receive 7 students BITSAT score and other scores (which are valid scores), and store these into Arrays. Generate MSNs for each and allot streams/ branches based on their BITSAT scores and preferences.

**Program:**

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4
5  int main() {
6  int n = 7;
7  int bitsat[14], boards_phy[n], boards_chem[n], boards_math[n], appln[n];
8  int passyr[n], result[n];
9  int seats[] = {2, 2, 2};
10 int preference[3 * n];
11     // 0 means CS, 1 means Civil, 2 means EEE, 3 Not Allotted
12 int eligibility[7] = {};
13     for (int i = 0; i < 7; i++) {
14 printf("Please Enter Student %d's application number: ", i + 1);
15 scanf("%d", &appln[i]);
16 printf("Please Enter Student %d's Physics score in boards: ", i + 1);
17 scanf("%d", &boards_phy[i]);
18 printf("Please Enter Student %d's Chemistry score in boards: ", i + 1);
19 scanf("%d", &boards_chem[i]);
20 printf("Please Enter Student %d's Maths score in boards: ", i + 1);
21 scanf("%d", &boards_math[i]);
22 printf("Please Enter Student %d's Board passing year: ", i + 1);
23 scanf("%d", &passyr[i]);
24 printf("Please Enter Student %d's BITSAT score: ", i + 1);
25 scanf("%d", &bitsat[2 * i]);
26 bitsat[2 * i + 1] = i; //2i+1 is index
27 printf("Please Enter Student %d's Preference ", i + 1);
28     for (int j = 3 * i; j < 3 * (i + 1); j++) {
29 scanf("%d", &preference[j]);
30     }
31 printf("\n");
32     }
33     for (int i = 0; i < 7; i++) {
34 printf("\n");
35     if (boards_phy[i] >= 60 && boards_chem[i] >= 60 && boards_math[i] >= 60) {
36         if (boards_phy[i] + boards_chem[i] + boards_math[i] >= 225) {
37             if (bitsat[2 * i] >= 240) {
38                 if (passyr[i] == 2021 || passyr[i] == 2020)

```



```

39         eligibility[i] = 1;
40     }
41 }
42 }
43 }
44 for (int i = 0; i < 14; i += 2) {
45     for (int j = i + 2; j < 14; j += 2) {
46         if (bitsat[i] < bitsat[j]) {
47             int tempMark = bitsat[i], tempIndex = bitsat[i + 1];
48             bitsat[i] = bitsat[j];
49             bitsat[i + 1] = bitsat[j + 1];
50             bitsat[j] = tempMark;
51             bitsat[j + 1] = tempIndex;
52         }
53     }
54 }
55 for (int i = 0; i < 7; i++) {
56     int index = bitsat[2 * i + 1];
57     if (eligibility[index] == 0)
58         result[index] = 3;
59     else {
60         int pref1 = preference[3 * index];
61         int pref2 = preference[(3 * index) + 1];
62         int pref3 = preference[(3 * index) + 2];
63         if (seats[pref1] != 0) {
64             result[index] = pref1;
65             seats[pref1] = seats[pref1] - 1;
66         }
67         else if (seats[pref2] != 0) {
68             result[index] = pref2;
69             seats[pref2] = seats[pref2] - 1;
70         }
71         else if (seats[pref3] != 0) {
72             result[index] = pref3;
73             seats[pref3] = seats[pref3] - 1;
74         }
75         else
76             result[index] = 3;

```

```

77         }
78     }
79     printf("Allotment: \n");
80     for (int i = 0; i < 7; i++){
81         printf("Student%d: Application no.: %d ", i + 1, appln[i]);
82         switch (result[i]) {
83             case 0:
84                 printf("CS");
85                 break;
86             case 1:
87                 printf("Civil");
88                 break;
89             case 2:
90                 printf("EEE");
91                 break;
92             case 3:
93                 printf("-");
94                 break;
95             }
96         printf("\n");
97     }
98 }

```

**Output:**

Please Enter Student 1's application number: 1001  
Please Enter Student 1's Physics score in boards: 89  
Please Enter Student 1's Chemistry score in boards: 96  
Please Enter Student 1's Maths score in boards: 79  
Please Enter Student 1's Board passing year: 2021  
Please Enter Student 1's BITSAT score: 357  
Please Enter Student 1's Preference 0 1 2

Please Enter Student 2's application number: 1002  
Please Enter Student 2's Physics score in boards: 86  
Please Enter Student 2's Chemistry score in boards: 76  
Please Enter Student 2's Maths score in boards: 90  
Please Enter Student 2's Board passing year: 2020  
Please Enter Student 2's BITSAT score: 378  
Please Enter Student 2's Preference 0 1 2

Please Enter Student 3's application number: 1003  
Please Enter Student 3's Physics score in boards: 89  
Please Enter Student 3's Chemistry score in boards: 76  
Please Enter Student 3's Maths score in boards: 68  
Please Enter Student 3's Board passing year: 2021  
Please Enter Student 3's BITSAT score: 320  
Please Enter Student 3's Preference 0 2 1

Please Enter Student 4's application number: 1004  
Please Enter Student 4's Physics score in boards: 88  
Please Enter Student 4's Chemistry score in boards: 77  
Please Enter Student 4's Maths score in boards: 99  
Please Enter Student 4's Board passing year: 2021  
Please Enter Student 4's BITSAT score: 335  
Please Enter Student 4's Preference 0 1 2

Please Enter Student 5's application number: 1005  
Please Enter Student 5's Physics score in boards: 88  
Please Enter Student 5's Chemistry score in boards: 86  
Please Enter Student 5's Maths score in boards: 97  
Please Enter Student 5's Board passing year: 2021  
Please Enter Student 5's BITSAT score: 376  
Please Enter Student 5's Preference 0 2 1

Please Enter Student 6's application number: 1006  
Please Enter Student 6's Physics score in boards: 89  
Please Enter Student 6's Chemistry score in boards: 97  
Please Enter Student 6's Maths score in boards: 88  
Please Enter Student 6's Board passing year: 2020  
Please Enter Student 6's BITSAT score: 320  
Please Enter Student 6's Preference 1 2 0

Please Enter Student 7's application number: 1007  
Please Enter Student 7's Physics score in boards: 88  
Please Enter Student 7's Chemistry score in boards: 87  
Please Enter Student 7's Maths score in boards: 86  
Please Enter Student 7's Board passing year: 2020  
Please Enter Student 7's BITSAT score: 356  
Please Enter Student 7's Preference 0 2 1

#### Allotment:

Student1: Application no.: 1001 Civil  
Student2: Application no.: 1002 CS  
Student3: Application no.: 1003 -  
Student4: Application no.: 1004 Civil  
Student5: Application no.: 1005 CS  
Student6: Application no.: 1006 EEE  
Student7: Application no.: 1007 EEE

### Task:

1. Modify the program in Que. 2 to count the number of special characters additionally along with other output.
2. Extend the program in Que. 4 to allot 15 students competing for 3 seats each in CSE, Civil and EEE departments.
3. Write a C program to print a  $[a \times a]$  matrix, for a given order  $a$ , which consists of numbers from 1 to  $a^2$  and each number must occur exactly once. The sum of elements in the row, column and the diagonal must be the same. Read the order of the matrix using the 'order' variable. Use a conditional statement to check whether the order is odd or even. If it is found to be even exit the program.

----- # -----