

1905

Pages: 5

Total Points 15

Duration: 45 minutes

## ESC101: Fundamentals of Computing(Major Quiz 2)

## Instructions:

1. Write you name, section and roll number on all the pages of the answer book.
2. Write the answers cleanly in the space provided. There is space left on the back of the answer book for rough work.
3. Using pens (blue/black ink) and not pencils. Do not use red pens for answering.
4. Even if no answers are written, the answer book has to be returned back with name and roll number written.
5. Recall that cheating carries severe consequences.

Question	Points	Score
1	10	
2	5	
Total:	15	

Name:	DIVYANSH SINGHVI
Roll No:	150238
Section:	A-4

Name: DIVYANSH SINGHVI Section: A-4 Rollno: 150238

Question 1. (10 points) Consider the program given below.

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 typedef struct str
5 {
6     int a, b;
7     int id;
8 }A;
9
10 void foo(A *arr, int len, int num, int cur, int wa, int wb, int sum)
11 {
12     if( num == len){
13         printf("%d\n",sum);
14         return;
15     }
16     sum = arr[cur].a*wa+arr[cur].b*wb;
17     printf("%d ",arr[cur].id );
18     foo(arr, len, num+1, (cur+1)%len, wa-2, wb-2, sum);
19 }
20
21 int main()
22 {
23     int len, i;
24     A *arr;
25     int wa, wb;
26     scanf("%d", &len);
27     arr = (A*) malloc(sizeof(A) * len);
28     for( i=0; i<len; i++)
29         scanf("%d %d %d",&arr[i].id, &arr[i].a, &arr[i].b);
30     scanf("%d %d",&wa, &wb);
31     for( i=0; i<len; i++)
32         foo(arr, len, 0, i, wa, wb, 0);
33     free(arr);
34     return 0;
35 }

```

What is the output of the program? If the program results in an error, mention the type of error.

Input: 3 1 2 3 2 3 4 3 4 5 10 20  
Output:

1	2	3	104	✓
2	3	1	60	✓
3	1	2	82	✓

Input: 2 1 3 5 2 4 9 12 7  
Output:

1	2	85	✓
2	1	55	✓

wa = 10, wb = 20

id	a	b
1	2	3
2	3	4
3	4	5

Page 2 of 5

1	2	3	104
2	3	1	60
3	1	2	82



Name: DIVYANSH SINGHVI Section: A-4

Rollno: 95023

Question 2. (5 points) Consider the program given below.

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 void move(int **mat, int x, int y, int r, int c, int t){
4     int distx, disty;
5     mat[y][x] = 1;
6     if(x == c && y == r)
7         return;
8     else{
9         distx = c-x; disty = r-y;
10        if(distx > disty){
11            if(mat[x+1][y] != 2)
12                move(mat, x+1, y, r, c, t+1);
13            else if(mat[x][y+1] != 2)
14                move(mat, x, y+1, r, c, t+1);
15            else
16                move(mat, x+1, y+1, r, c, t+1);
17        }
18        else{
19            if(mat[x][y+1] != 2)
20                move(mat, x, y+1, r, c, t+1);
21            else if(mat[x+1][y] != 2)
22                move(mat, x+1, y, r, c, t+1);
23            else
24                move(mat, x+1, y+1, r, c, t+1);
25        }
26    }
27 }
28 int main(){
29     int n, nm;
30     int r, c, i, j;
31     int tx, ty;
32     int **mat;
33     scanf("%d", &n);
34     scanf("%d %d", &r, &c);
35     mat = (int**) malloc(sizeof(int*) * n);
36     scanf("%d", &nm);
37     for(i=0; i<n; i++){
38         mat[i] = (int*) malloc(sizeof(int) * n);
39         for(j=0; j<n; j++){
40             mat[i][j]=0;
41         }
42     }
43     for(i=0; i<nm; i++){
44         scanf("%d %d", &tx, &ty);
45         mat[ty][tx] = 2;
46     }
47     move(mat, 0,0, r, c, 0);
48     for(i=0; i<n; i++){
49         for(j=0; j<n; j++){
50             printf("%d ", mat[i][j]);
51         }
52         printf("\n");
53     }
54 }

```

xy  
(1, 0, 1, 2)  
(2, 0, 1, 2)  
(2, 1, 1, 2)

t=2

all zeros  
mat

1	1	1
0	2	1
0	0	0

n=4  
r=2  
c=2  
nm=2  
tx 1 2  
ty 2 1

(1, 1, 2, 2)  
(2, 1, 2, 2)  
(2, 2, 2, 2)

1	0	0
1	1	2
0	2	1
0	0	0

0,2  
(1,2)  
(2,2)  
(2,2)

1			
1	1	1	
	1	1	

0,2) (0,1)  
(1,2) (0,1)  
(2,2) (1,1)

1	1	1
0	2	1
0	0	0

Name: DIVYANSH SINGHVI

Section: A-4

Rollno: 150238

```

51 printf("\n");
52 }
53 for(i = 0; i < n; i++)
54     free(mat[i]);
55 free(mat);
56 }
    
```

What is the output of the program? If the program results in an error, mention the type of error.

Input: 3 1 2 1 1 1

Output:

1	1	1	✓
0	2	1	
0	0	0	

2/5

1 1 1  
0 2 1  
0 0 0

Input: 4 2 2 2 1 2 2 1

Output:

1	0	0	0	
1	1	1	0	
0	1	1	0	X
0	0	0	0	