

Q1

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5 int main(int argc, char *argv[]) {
6     return 0; #include <stdio.h>
7     #include <stdlib.h>
8
9 int main() {
10     int n, add, threshold;
11     printf("Hours: ");
12     scanf("%d", &n);
13
14     int *steps = malloc(n * sizeof(int));
15
16     for (int i = 0; i < n; i++)
17         scanf("%d", &steps[i]);
18
19     printf("More hours: ");
20     scanf("%d", &add);
21
22     steps = realloc(steps, (n + add) * sizeof(int));
23
24     for (int i = n; i < n + add; i++)
25         scanf("%d", &steps[i]);
26
27     n += add;
28
29     printf("Threshold: ");
30     scanf("%d", &threshold);
31
32     int max = steps[0], total = 0, above = 0;
33     for (int i = 0; i < n; i++) {
34         if (steps[i] > max) max = steps[i];
35         if (steps[i] > threshold) above++;
36         total += steps[i];
37     }
38
39     FILE *f = fopen("fitness_tracker.txt", "w");
40     fprintf(f, "%d %d %d", max, total, above);
41     fclose(f);
42
43     free(steps);
44 }
45
```

The image shows a screenshot of a C program being executed in a debugger. The top panel shows the source code in `main.c` with a line number 1 and the code `9 32 0`. The bottom panel shows the program's output in a console window. The output consists of several lines of text: `Hours: 5`, `6`, `5`, `4`, `3`, `2`, `More hours: 3`, `2`, `1`, `9`, and `Threshold: 87`. At the end of the output, it says `...Program finished with exit code 0` and `Press ENTER to exit console.` with a cursor.

```
main.c fitness_tracker.txt
1 9 32 0

Hours: 5
6
5
4
3
2
More hours: 3
2
1
9
Threshold: 87

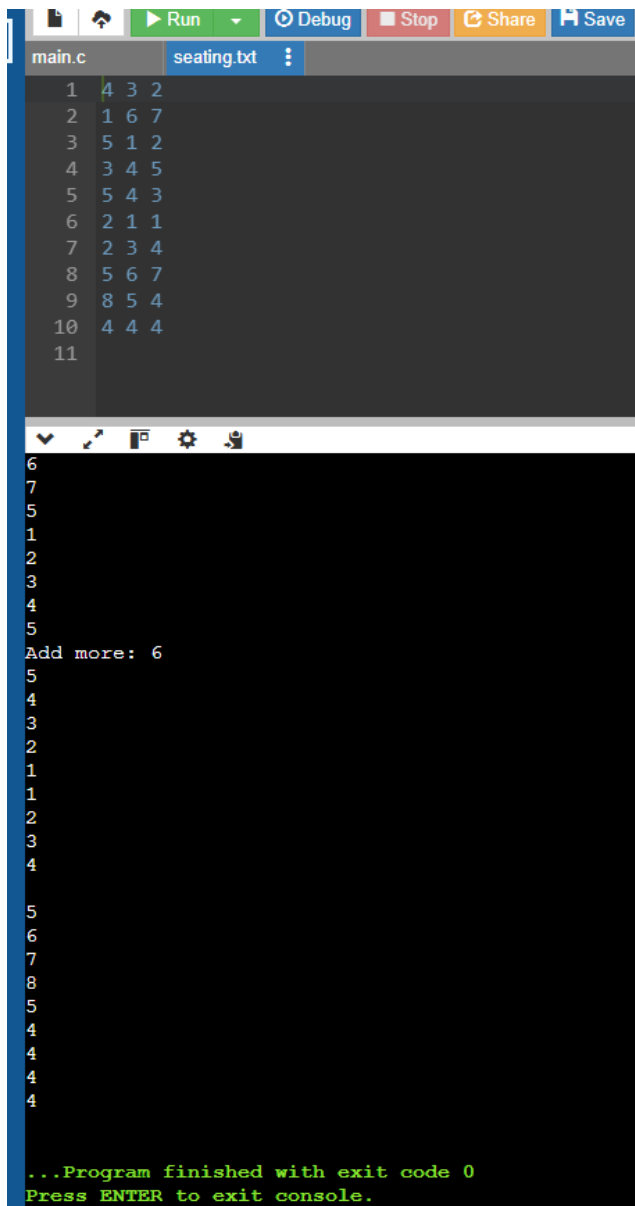
...Program finished with exit code 0
Press ENTER to exit console.
```

Q2

```

1
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <ctype.h>
5
6 int check(char *s, int i) {
7     if (s[i] == '\0') return 1;
8     if (!isalnum(s[i])) return 0;
9     return check(s, i+1);
10 }
11
12 typedef struct {
13     char name[50];
14     char roll[20];
15     int seat;
16 } Seat;
17
18 int main() {
19     int n, add;
20     printf("Students: ");
21     scanf("%d", &n);
22
23     Seat *s = malloc(n * sizeof(Seat));
24
25     for (int i = 0; i < n; i++) {
26         scanf("%s %s %d", s[i].name, s[i].roll, &s[i].seat);
27         while (!check(s[i].roll, 0))
28             scanf("%s", s[i].roll);
29     }
30
31     printf("Add more: ");
32     scanf("%d", &add);
33
34     s = realloc(s, (n + add) * sizeof(Seat));
35
36     for (int i = n; i < n + add; i++) {
37         scanf("%s %s %d", s[i].name, s[i].roll, &s[i].seat);
38         while (!check(s[i].roll, 0))
39             scanf("%s", s[i].roll);
40     }
41
42     FILE *f = fopen("seating.txt", "w");
43     for (int i = 0; i < n + add; i++)
44         fprintf(f, "%s %s %d\n", s[i].name, s[i].roll, s[i].seat);
45
46     fclose(f);
47     free(s);
48 }
49

```



The image shows a code editor window with two tabs: 'main.c' and 'seating.txt'. The 'seating.txt' tab is active, displaying a list of 11 rows of numbers. Below the code editor is a console window showing the output of the program. The output consists of 11 rows of numbers, followed by the prompt 'Add more: 6', and then another 11 rows of numbers. The program ends with the message '...Program finished with exit code 0' and 'Press ENTER to exit console.'

```
main.c  seating.txt  ...
1  4 3 2
2  1 6 7
3  5 1 2
4  3 4 5
5  5 4 3
6  2 1 1
7  2 3 4
8  5 6 7
9  8 5 4
10 4 4 4
11
6
7
5
1
2
3
4
5
Add more: 6
5
4
3
2
1
1
2
3
4
5
6
7
8
5
4
4
4
4
...Program finished with exit code 0
Press ENTER to exit console.
```

Q3

main.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      int n, add, th;
6      printf("Readings: ");
7      scanf("%d", &n);
8
9      float *t = malloc(n * sizeof(float));
10
11     for (int i = 0; i < n; i++)
12         scanf("%f", &t[i]);
13
14     printf("Add: ");
15     scanf("%d", &add);
16
17     t = realloc(t, (n + add) * sizeof(float));
18
19     for (int i = n; i < n + add; i++)
20         scanf("%f", &t[i]);
21
22     n += add;
23
24     printf("Alert threshold: ");
25     scanf("%d", &th);
26
27     float max = t[0], min = t[0];
28     int count = 0;
29
30     for (int i = 0; i < n; i++) {
31         if (t[i] > max) max = t[i];
32         if (t[i] < min) min = t[i];
33         if (t[i] > th) count++;
34     }
35
36     FILE *f = fopen("temperature_summary.txt", "w");
37     fprintf(f, "%.2f %.2f %d", max, min, count);
38     fclose(f);
39
40     free(t);
41 }
```

```
main.c  temperature_sum...
1 7.00 2.00 0

Readings: 4
3
5
6
7
Add: 4
3
2
7
6
Alert threshold: 8

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

Q4

```
main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 typedef struct {
5     char title[50];
6     char id[20];
7     char time[20];
8     int due;
9 } Log;
10
11 int main() {
12     int n;
13     scanf("%d", &n);
14
15     Log *a = malloc(n * sizeof(Log));
16     FILE *f = fopen("checkout_log.csv", "a");
17
18     for (int i = 0; i < n; i++) {
19         scanf("%s %s %s %d", a[i].title, a[i].id, a[i].time, &a[i].due);
20         fprintf(f, "%s,%s,%s,%d\n", a[i].title, a[i].id, a[i].time, a[i].due);
21     }
22
23     fclose(f);
24     free(a);
25 }
26
```

Q5

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

int main() {
    float rate;
    printf("Rate: ");
    scanf("%f", &rate);

    int n;
    printf("Rentals: ");
    scanf("%d", &n);

    float *d = malloc(n * sizeof(float));

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

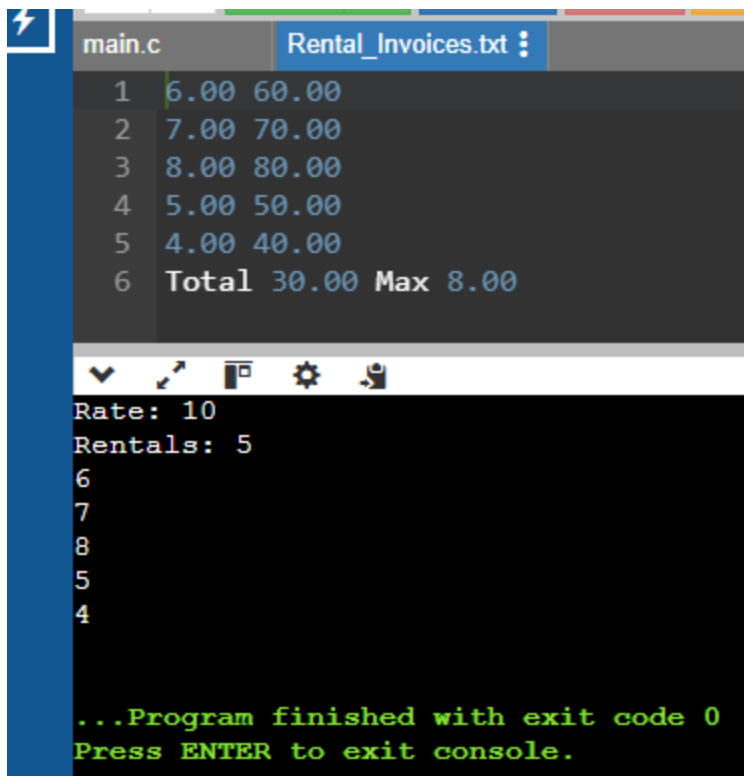
    float total = 0, max = d[0];

    FILE *f = fopen("Rental_Invoices.txt", "w");

    for (int i = 0; i < n; i++) {
        float cost = d[i] * rate;
        fprintf(f, "%.2f %.2f\n", d[i], cost);
        total += d[i];
        if (d[i] > max) max = d[i];
    }

    fprintf(f, "Total %.2f Max %.2f", total, max);
    fclose(f);

    free(d);
}
```



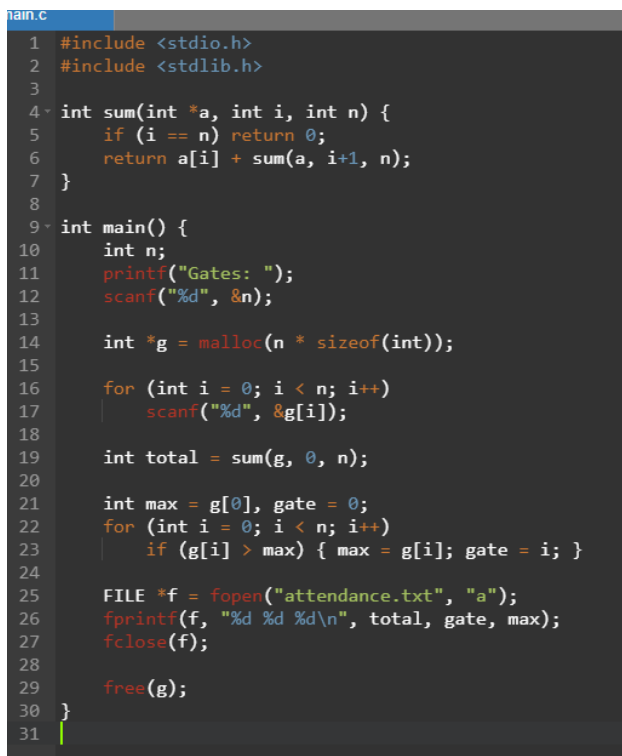
The screenshot shows a code editor with two tabs: 'main.c' and 'Rental_Invoices.txt'. The 'main.c' tab is active, displaying a C program that calculates the sum of an array and finds the maximum value. The output of the program is shown in the terminal window below the editor. The output displays the rate (10), the number of rentals (5), and the total sum (30.00) and maximum value (8.00). The program finishes with exit code 0.

```
main.c Rental_Invoices.txt :
1 6.00 60.00
2 7.00 70.00
3 8.00 80.00
4 5.00 50.00
5 4.00 40.00
6 Total 30.00 Max 8.00

Rate: 10
Rentals: 5
6
7
8
5
4

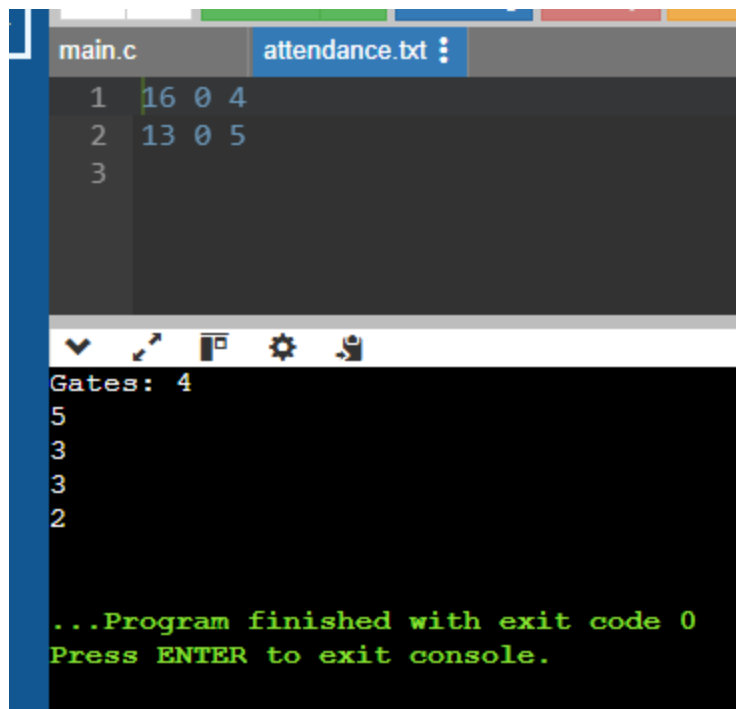
...Program finished with exit code 0
Press ENTER to exit console.
```

Q6



The screenshot shows the source code of the C program in the 'main.c' tab. The code includes standard headers, defines a recursive sum function, and a main function that reads input, calculates the sum and maximum, and writes the results to a file named 'attendance.txt'.

```
main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int sum(int *a, int i, int n) {
5     if (i == n) return 0;
6     return a[i] + sum(a, i+1, n);
7 }
8
9 int main() {
10     int n;
11     printf("Gates: ");
12     scanf("%d", &n);
13
14     int *g = malloc(n * sizeof(int));
15
16     for (int i = 0; i < n; i++)
17         scanf("%d", &g[i]);
18
19     int total = sum(g, 0, n);
20
21     int max = g[0], gate = 0;
22     for (int i = 0; i < n; i++)
23         if (g[i] > max) { max = g[i]; gate = i; }
24
25     FILE *f = fopen("attendance.txt", "a");
26     fprintf(f, "%d %d %d\n", total, gate, max);
27     fclose(f);
28
29     free(g);
30 }
31
```

The image shows a code editor window with two tabs: 'main.c' and 'attendance.txt'. The 'attendance.txt' tab is active, displaying three lines of text: '1 16 0 4', '2 13 0 5', and '3'. Below the code editor is a terminal window. The terminal output shows 'Gates: 4' followed by the numbers '5', '3', '3', and '2' on separate lines. At the bottom of the terminal, a green message states: '...Program finished with exit code 0' and 'Press ENTER to exit console.'

```
main.c  attendance.txt :
1 16 0 4
2 13 0 5
3

Gates: 4
5
3
3
2

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

Q7

```

main.c
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct {
5      char name[50];
6      int qty;
7      float price;
8  } Med;
9
10 int main() {
11     int n;
12     scanf("%d", &n);
13
14     Med *m = malloc(n * sizeof(Med));
15
16     for (int i = 0; i < n; i++)
17         scanf("%s %d %f", m[i].name, &m[i].qty, &m[i].price);
18
19     float total = 0;
20
21     FILE *f = fopen("medicine_inventory.txt", "w");
22
23     for (int i = 0; i < n; i++) {
24         fprintf(f, "%s %d %.2f\n", m[i].name, m[i].qty, m[i].price);
25         total += m[i].qty * m[i].price;
26     }
27
28     fprintf(f, "Total %.2f", total);
29     fclose(f);
30
31     free(m);
32 }
33

```

```

main.c  medicine_inventor...
1  4 2 1.00
2  3 4 5.00
3  6 5 4.00
4  Total 42.00

3
4
2
1
3
4
5
6
5
4

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

```

