

Algorithm - assignment 2

물리학과 20182326 이선민

Problem 1

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int k = 1;
6     int num = 2;
7     int n;
8     printf("Enter a number: ");
9     scanf("%d", &n);
10    while (num <= n)
11    {
12        num *= 2;
13        k++;
14    }
15    if (num > n)
16        k--;
17    printf("k: %d\n", k);
18 }
```

```
[~/2022_1/Algorithm/assignment2]$ gcc problem1.c
[~/2022_1/Algorithm/assignment2]$ ./a.out
Enter a number: 10
k: 3
[~/2022_1/Algorithm/assignment2]$ ./a.out
Enter a number: 50
k: 5
[~/2022_1/Algorithm/assignment2]$ ./a.out
Enter a number: 1025
k: 10
```

Problem 2

```
1 #include <stdio.h>
2
3 int is_palindrome(char *str)
4 {
5     int str_len = 0;
6     while(str[str_len])
7         str_len++;
8     for (int i = 0; i < (str_len/2); i++)
9     {
10         if (str[i] != str[str_len - 1 - i])
11             return (0);
12     }
13     return (1);
14 }
15
16 int main()
17 {
18     char str1[10] = "hello";
19     char str2[10] = "radar";
20
21     if (is_palindrome(str1))
22         printf("%s is palindrome\n", str1);
23     else
24         printf("%s is not palindrome\n", str1);
25     if (is_palindrome(str2))
26         printf("%s is palindrome\n", str2);
27     else
28         printf("%s is not palindrome\n", str2);
29 }
```

```
[~/2022_1/Algorithm/assignment2]$ gcc problem2.c
[~/2022_1/Algorithm/assignment2]$ ./a.out
hello is not palindrome
radar is palindrome
```

Problem 3

problem 3

$$\underbrace{((x > y) ? x : y)}_x \geq z ? \underbrace{((y > x) ? x : y)}_y : z$$

$$\Rightarrow x > z ? y : z = z$$

5

Problem 4

```
1 #include <stdio.h>
2
3 void print_matrix(int mat[5][5], int row, int col)
4 {
5     int tmp;
6     int count;
7     for (int i = 0; i < row; i++)
8     {
9         for (int j = 0; j < col; j++)
10        {
11            tmp = mat[i][j];
12            count = 0;
13            while (tmp > 0)
14            {
15                count++;
16                tmp /= 10;
17            }
18            for (int k = 0; k < (3 - count); k++)
19                printf(" ");
20            printf("%d ", mat[i][j]);
21        }
22        printf("\n");
23    }
24 }
25
```

```
1 #include <stdio.h>
2
3 int binary_search(int *arr, int find, int start, int end)
4 {
5     int mid = (end + start)/2;
6     if (end == start)
7         return -1;
8     if (find == arr[mid])
9         return (mid);
10    else if (find > arr[mid])
11        return binary_search(arr, find, mid + 1, end);
12    else if (find < arr[mid])
13        return binary_search(arr, find, start, mid - 1);
14    return (-1);
15 }
16
17 int main()
18 {
19     int index;
20     int arr[9] = {12, 34, 37, 45, 57, 82, 99, 120, 134};
21     int to_find = 120;
22     index = binary_search(arr, to_find, 0, 8);
23     if (index != -1)
24         printf("%d is located in Array index %d.\n", to_find, index);
25     else
26         printf("Number is not in array.\n");
27 }
~
```

Problem 4 - result

```
[~/2022_1/Algorithm/assignment2]$ gcc problem4.c  
[~/2022_1/Algorithm/assignment2]$ ./a.out  
120 is located in Array index 7.
```

Problem 5

```
26 void rotate_matrix(int mat[5][5], int row, int col)
27 {
28     int i;
29     int j;
30     int tmp[5][5];
31     for (i = 0; i < row; i++)
32     {
33         for (j = 0; j < col; j++)
34             tmp[i][j] = mat[4 - j][i];
35     }
36     for (i = 0; i < row; i++)
37     {
38         for (j = 0; j < col; j++)
39             mat[i][j] = tmp[i][j];
40     }
41 }
42
43 int main()
44 {
45     int matrix[5][5] = {{1,2,3,4,5},{6,7,8,9,10},{11,12,13,14,15},
46                        {16,17,18,19,20},{21,22,23,24,25}};
47     printf("original array\n");
48     print_matrix(matrix, 5, 5);
49
50     rotate_matrix(matrix, 5, 5);
51     printf("\nrotate array\n");
52     print_matrix(matrix, 5, 5);
53
54     rotate_matrix(matrix, 5, 5);
55     printf("\nsecond rotate array\n");
56     print_matrix(matrix, 5, 5);
57 }
58
```

```
[~/2022_1/Algorithm/assignment2]$ gcc problem5.c
[~/2022_1/Algorithm/assignment2]$ ./a.out
```

original array

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

rotate array

21	16	11	6	1
22	17	12	7	2
23	18	13	8	3
24	19	14	9	4
25	20	15	10	5

second rotate array

25	24	23	22	21
20	19	18	17	16
15	14	13	12	11
10	9	8	7	6
5	4	3	2	1

Problem 6

```
1 #include <stdio.h>
2
3 void pairSum(int *arr, int sum, int arr_size)
4 {
5     int i;
6     int j;
7     int output_size = 0;
8     int k = 0;
9
10    for (i = 0; i < arr_size; i++)
11        for (j = i + 1; j < arr_size; j++)
12            if (arr[i] + arr[j] == sum)
13                output_size++;
14
15    printf("[");
16    for (i = 0; i < arr_size; i++)
17    {
18        for (j = i + 1; j < arr_size; j++)
19        {
20            if (arr[i] + arr[j] == sum)
21            {
22                printf("%d+%d", arr[i], arr[j]);
23                k++;
24                if (k != output_size)
25                    printf(",");
26            }
27        }
28        printf("]\n");
29    }
30
31 int main()
32 {
33     int arr[10] = {2, 4, 3, 5, 6, -2, 4, 7, 8, 9};
34     pairSum(arr, 7, 10);
35 }
```

```
[~/2022_1/Algorithm/assignment2]$ gcc problem6.c
[~/2022_1/Algorithm/assignment2]$ ./a.out
['2+5', '4+3', '3+4', '-2+9']
```


Problem 7

```
1 #include <stdio.h>
2
3 int *ordinary_mul(int first[][50], int second[][50], int first_row, int first_col, int sec_row, int sec_col)
4 {
5     static int ans[50][50];
6     int tmp;
7
8     int i, j, h;
9     for (i = 0; i < first_row; i++)
10    {
11        for (h = 0; h < sec_col; h++)
12        {
13            tmp = 0;
14            for (j = 0; j < first_col; j++)
15            {
16                tmp += first[i][j] * second[j][h];
17            }
18            ans[i][h] = tmp;
19        }
20    }
21    return (int *)ans;
22 }
23
24 int *arr_plus(int first[][50], int second[][50], int row, int col)
25 {
26     static int ans[50][50];
27     int tmp;
28
29     int i, j;
30     for (i = 0; i < row; i++)
31     {
32         for (j = 0; j < col; j++)
33         {
34             ans[i][j] = first[i][j] + second[i][j];
35         }
36     }
37     return (int *)ans;
38 }
39
```

Problem 7

```
39
40 int *arr_minus(int first[][50], int second[][50], int row, int col)
41 {
42     static int ans[50][50];
43     int tmp;
44
45     int i, j;
46     for (i = 0; i < row; i++)
47     {
48         for (j = 0; j < col; j++)
49         {
50             ans[i][j] = first[i][j] - second[i][j];
51         }
52     }
53     return (int *)ans;
54 }
55
```

Problem 7

```
56 int *strassen_mul(int first[][50], int second[][50], int first_row, int first_col, int sec_col)
57 {
58     static int ans[50][50];
59     int i, j;
60     // first_row = 30, first_col = second_row = 10, sec_col = 50
61     int first11[50][50];
62     int first12[50][50];
63     int first21[50][50];
64     int first22[50][50];
65     int second11[50][50];
66     int second12[50][50];
67     int second21[50][50];
68     int second22[50][50];
69     for (i = 0; i < 15; i++)
70     {
71         for (j = 0; j < 5; j++)
72         {
73             first11[i][j] = first[i][j];
74             first12[i][j] = first[i][j+5];
75             first21[i][j] = first[i+15][j];
76             first22[i][j] = first[i+15][j + 5];
77         }
78     }
79     for (i = 0; i < 5; i++)
80     {
81         for (j = 0; j < 25; j++)
82         {
83             second11[i][j] = second[i][j];
84             second12[i][j] = second[i][j+25];
85             second21[i][j] = second[i+5][j];
86             second22[i][j] = second[i+5][j+25];
87         }
88     }
89 }
```

Problem 7

```
90
91     int (*P1)[50] = (void *)ordinary_mul(first11, (void *)arr_minus(second12, second22,5,25),15,5,5,25);
92     int (*P2)[50] = (void *)ordinary_mul((void *)arr_plus(first11, first12, 15,5), second22,15,5,5,25);
93     int (*P3)[50] = (void *)ordinary_mul((void *)arr_plus(first21, first22,15,5), second11,15,5,5,25);
94     int (*P4)[50] = (void *)ordinary_mul(first22, (void *)arr_minus(second21, second11,5,25),15,5,5,25);
95     int (*P5)[50] = (void *)ordinary_mul((void *)arr_plus(first11, first22,15,5), (void *)arr_plus(second11, second22,5,25),15,5,5,25);
96     int (*P6)[50] = (void *)ordinary_mul((void *)arr_minus(first12,first22,15,5), (void *)arr_plus(second21, second22,5,25),15,5,5,25);
97     int (*P7)[50] = (void *)ordinary_mul((void *)arr_minus(first11,first21,15,5), (void *)arr_plus(second11, second12,5,25),15,5,5,25);
98     int (*R)[50] = (void *)arr_minus((void *)arr_plus((void *)arr_plus(P5,P4,15,25),P6,15,25),P2, 15,25);
99     int (*S)[50] = (void *)arr_plus(P1, P2, 15,25);
100    int (*T)[50] = (void *)arr_plus(P3, P4, 15,25);
101    int (*U)[50] = (void *)arr_minus((void *)arr_minus((void *)arr_plus(P5,P1,15,25),P3,15,25),P7, 15,25);
102
103
104    for (i = 0; i < 15; i++)
105    {
106        for (j = 0; j < 25; j++)
107        {
108            ans[i][j] = R[i][j];
109            ans[i][j+25] = S[i][j];
110            ans[i+15][j] = T[i][j];
111            ans[i+15][j+25] = U[i][j];
112        }
113    }
114    return (int *)ans;
115 }
116
```

Problem 7

```
117 int main()
118 {
119     int first_arr[50][50] = {
120         {1,1,1,1,1,1,1,1,1,1},
121         {1,1,1,1,1,1,1,1,1,1},
122         {1,1,1,1,1,1,1,1,1,1},
123         {1,1,1,1,1,1,1,1,1,1},
124         {1,1,1,1,1,1,1,1,1,1},
125         {1,1,1,1,1,1,1,1,1,1},
126         {1,1,1,1,1,1,1,1,1,1},
127         {1,1,1,1,1,1,1,1,1,1},
128         {1,1,1,1,1,1,1,1,1,1},
129         {1,1,1,1,1,1,1,1,1,1},
130         {1,1,1,1,1,1,1,1,1,1},
131         {1,1,1,1,1,1,1,1,1,1},
132         {1,1,1,1,1,1,1,1,1,1},
133         {1,1,1,1,1,1,1,1,1,1},
134         {1,1,1,1,1,1,1,1,1,1},
135         {1,1,1,1,1,1,1,1,1,1},
136         {1,1,1,1,1,1,1,1,1,1},
137         {1,1,1,1,1,1,1,1,1,1},
138         {1,1,1,1,1,1,1,1,1,1},
139         {1,1,1,1,1,1,1,1,1,1},
140         {1,1,1,1,1,1,1,1,1,1},
141         {1,1,1,1,1,1,1,1,1,1},
142         {1,1,1,1,1,1,1,1,1,1},
143         {1,1,1,1,1,1,1,1,1,1},
144         {1,1,1,1,1,1,1,1,1,1},
145         {1,1,1,1,1,1,1,1,1,1},
146         {1,1,1,1,1,1,1,1,1,1},
147         {1,1,1,1,1,1,1,1,1,1},
148         {1,1,1,1,1,1,1,1,1,1},
149         {1,1,1,1,1,1,1,1,1,1}
150     };
```

Problem 7

```
150     },
151     int second_arr[50][50] = {
152         {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
153         {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
154         {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
155         {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
156         {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
157         {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
158         {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
159         {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
160         {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2},
161         {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2}
162     };
163     printf("ordinary multiplication\n");
164     int (*arr)[50] = (void *)ordinary_mul(first_arr, second_arr, 30, 10, 10, 50);
165     for (int i = 0; i < 30; i++)
166     {
167         for (int j = 0; j < 50; j++)
168             printf("%d ", arr[i][j]);
169         printf("\n");
170     }
171
172     int (*please)[50] = (void *)strassen_mul(first_arr, second_arr, 30, 10, 50);
173     printf("\n\nstrassen multiplication\n");
174     for (int i = 0; i < 30; i++)
175     {
176         for (int j = 0; j < 50; j++)
177             printf("%d ", please[i][j]);
178         printf("\n");
179     }
180     //strassen_mul(first_arr, second_arr, 30, 10, 50);
181 }
```

Problem 7

```
[~/2022_1/Algorithm/assignment2]$ gcc problem7.c
```

```
[~/2022_1/Algorithm/assignment2]$ ./a.out
```

ordinary multiplication

[illegible]

Problem 7

strassen multiplication

[illegible]

Problem 7

- Runtime comparison failed because the Strassen method did not work properly.