1. 1248163264128
2. c
3.

```c
#include<stdio.h>

int main(void){
    int i;

    for (i=1;  i<=128; i*=2){
        printf("%d", i);
    }

    return 0;
}
```

4.

```c
#include<stdio.h>

int main(void){
    int n, num = 1;
    printf("TABLE OF POWERS OF TWO\n\n");
    printf(" n   2 to the n\n");
    printf("---  ------------\n");

    for (n=0; n<=10; n++){
        // printf("%d", num);
        num*=2;
        printf("\n%-2d   %-2d", n, num);
    }

    return 0;
}
```

5.

asc5.c > ...

```c
#include<stdio.h>

int main(){
    int i, days, first;

    printf("Enter number of days in month: ");
    scanf("%d", &days);

    switch(days){
        case 31: case 30: case 28: case 29:
            printf("Enter the starting day of the week (1=Sun, 7=Sat): ");
            scanf("%d", &first);
            switch(first){
                case 1: case 2: case 3: case 4: case 5: case 6: case 7:
                    for (i=1; i<first; i++){
                        printf("    ");
                    }
                    for(i=1; i<=days; i++){
                        printf("%3d", i);
                        if((i+first-1) % 7==0){
                            printf("\n");
                        }
                    }
                    break;
                default: printf("Invalid date"); break;
                }

            break;

        default: printf("Invalid date"); break;
    }

    return 0;
}
```

6a.

as6a.c > ...

```c
#include<stdio.h>
#include<stdbool.h>

#define NUM_PATHWAYS ((int)(sizeof(pathway)/sizeof(pathway[0])))

int main(){
    bool pathway[8]={[0]=true, [2]=true};
    for (int i=0; i<NUM_PATHWAYS; i++){
        if (pathway[i]){
            printf("pathway[%d] is open \n", i);
        }
        else{
            printf("pathway[%d] is close\n", i);
        }
    }

    return 0;
}

```

6b.

asc6b.c > ...

```c
#include<stdio.h>
#include<stdbool.h>

#define NUM_PATHWAYS ((int)(sizeof(pathway)/sizeof(pathway[0])))

int main(){
    int value;
    bool pathway[8]={false};
    for (value=0; value<=8; value++){
        if (value==0 || value==2){
            pathway[value]=true;
        }
    }

    for (int i=0; i<NUM_PATHWAYS; i++){
        if (pathway[i]){
            printf("pathway[%d] is open \n", i);
        }
        else{
            printf("pathway[%d] is close\n", i);
        }
    }

    return 0;
}

```

7.1

```c
bool road_networks[SIZE][SIZE]={
    {1,1,0,0,0,1,0,0},
    {1,1,1,0,0,0,0,0},
    {0,1,1,0,1,1,0,0},
    {0,0,0,1,1,0,0,0},
    {0,0,0,1,1,0,0,0},
    {1,0,1,0,0,1,0,0},
    {1,0,0,1,0,0,1,0},
    {0,0,0,0,0,1,0,1}
};
```

7.2

```
    //column
    char column[8]={'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'};

    for (int i=0; i<SIZE; i++){
        if (i==2||i==3){
            printf("   [%c]", column[i]);
        }
        else{
            printf("%6c", column[i]);
        }
    }
        printf("\n");

    //row
    char rowtitle[8]={'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'};

    //matrix
    for (int row=0; row<SIZE; row++){
        if (row==2||row==3){
            printf("[%c]  ", row[rowtitle]);
        }
        else{
            printf("%c    ", row[rowtitle]);
        }
        for (int column=0; column<SIZE; column++){
            printf ("%d     ", road_networks[row][column]);
        }
        printf("\n");
    }
```

7.3

```
    int point;
    printf("Which point are you located? 0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H: ");
    scanf("%d", &point);

    switch(point){
        case 0: printf("At point: A\npoint: C arrived to charging station"); break;
        case 1: printf("At point: B\npoint: C arrived to charging station"); break;
        case 5: printf("At point: F\npoint: C arrived to charging station"); break;

        case 4: printf("At point: E\npoint: D arrived to charging station"); break;
        case 6: printf("At point: G\npoint: D arrived to charging station"); break;
        case 7: printf("At point: H\npoint: D arrived to charging station"); break;

        case 2: printf("C is a charging station"); break;
        case 3: printf("D is a charging station"); break;

        default: printf("Invalid input"); break;
    }

return 0;
}
```

7.4

```c
#include<stdio.h>
#include<stdbool.h>
#define SIZE 8
```

Full Code:

```c
#include<stdio.h>
#include<stdbool.h>
#define SIZE 8

int main(){
    bool road_networks[SIZE][SIZE]={
        {1,1,0,0,0,1,0,0},
        {1,1,1,0,0,0,0,0},
        {0,1,1,0,1,1,0,0},
        {0,0,0,1,1,0,0,0},
        {0,0,0,1,1,0,0,0},
        {1,0,1,0,0,1,0,0},
        {1,0,0,1,0,0,1,0},
        {0,0,0,0,0,1,0,1}
    };

    //column
    char column[8]={'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'};

    for (int i=0; i<SIZE; i++){
        if (i==2||i==3){
            printf("   [%c]", column[i]);
        }
        else{
            printf("%6c", column[i]);
        }
    }
        printf("\n");

    //row
    char rowtitle[8]={'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'};

    //matrix
    for (int row=0; row<SIZE; row++){
        if (row==2||row==3){
            printf("[%c]  ", row[rowtitle]);
        }
        else{
            printf("%c    ", row[rowtitle]);
        }
```

```c
        for (int column=0; column<SIZE; column++){
            printf ("%d     ", road_networks[row][column]);
        }
        printf("\n");
    }

    int point;
    printf("Which point are you located? 0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H: ");
    scanf("%d", &point);

    switch(point){
        case 0: printf("At point: A\npoint: C arrived to charging station"); break;
        case 1: printf("At point: B\npoint: C arrived to charging station"); break;
        case 5: printf("At point: F\npoint: C arrived to charging station"); break;

        case 4: printf("At point: E\npoint: D arrived to charging station"); break;
        case 6: printf("At point: G\npoint: D arrived to charging station"); break;
        case 7: printf("At point: H\npoint: D arrived to charging station"); break;

        case 2: printf("C is a charging station"); break;
        case 3: printf("D is a charging station"); break;

        default: printf("Invalid input"); break;
    }

return 0;
}
```