



10 Libraries

[Random Number Generation](#)

[Home Work 1](#)

[Date Time Clock](#)

[Home Work 2](#)

10 Libraries

[C Library Reference](#)

Random Number Generation

[Random Number Generation Reference](#)

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

int main(void)
{
    srand(time(NULL)); // use current time as seed for random generator
    int random_variable = rand();
    printf("Random value on [0,%d]: %d\n", RAND_MAX, random_variable);

    int x;
    int count[] = { 0,0,0,0,0,0};
```

```

for (int i=0; i < 6000; i++) {
    x = 1+ rand()%6;
    //    printf("%d ", x);
    count[x-1]++;
}
printf("%d %d %d %d %d %d\n", count[0], count[1], count[2], count[3], count[4], count[5]);
}

```

Home Work 1

Write a program to generate and print:

- a uniformly random subset of {1, 2, ..., n}. Take n as input.
- a uniformly random subset of {1, 2, ..., n} of size k. Take n, k as input.
- a uniformly random permutation of {1, 2, ..., n}. Take n as input.

Date Time Clock

Date Time Clock Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include <time.h>

```

```

typedef enum RelStatus {
    NotMentioned,
    Single,
    Engaged,

```

```

    Married
} RelStatus;

typedef struct Person {
    char name[100];
    int age;
    RelStatus relstatus;
    int count_friends;
    struct Person* friends[5];
} Person;

typedef struct SocialNet {
    struct Person members[100];
    int size;
} SocialNet;

void print_person(struct Person p) {
    char status_string[][20] = {
        "Not Mentioned",
        "Single",
        "Engaged",
        "Married"
    };
    printf("%s\t\t%d\t%s\t\t", p.name, p.age, status_string[p.relstatus]);
    for (int i = 0; i < p.count_friends; i++) {
        printf("%s, ", p.friends[i]->name);
    }
    printf("\n");
}

void print_network(SocialNet social_net) {
    printf(
        "-----\n"

```

```

        "Name\t\tAge \t Rel Status\t\t\tFriends\n"
        "-----\n");
for (int i=0;i <social_net.size; i++) {
    print_person(social_net.members[i]);
}
printf("-----\n");

}

// Person* find_person(char* name1, SocialNet *sn) {
//     // TODO
// }

int main()
{

    clock_t now = clock();
    SocialNet social_net = {
        {
            {"Alice", 24, NotMentioned},
            {"Bob", 22, Married},
            {"Charlie", 28, Engaged}
        },
        3
    };

    social_net.members[0].friends[0] = &(social_net.members[1]);
    social_net.members[0].friends[1] = &(social_net.members[2]);
    social_net.members[0].count_friends = 2;
    social_net.members[2].friends[0] = &(social_net.members[1]);
    social_net.members[2].count_friends = 1;
    social_net.members[1].count_friends = 0;

```

```

    print_network(social_net);

    clock_t later = clock();

    printf("%d %d %f\n", later, now, ((float)(later-now) / CLOCKS_PER_SEC));

    return 0;
}

```

Home Work 2

- Write a program to print the number of days in the current month.
- Write a program, which takes an input a date and prints the month calendar where that date belongs.

For eg: For today date it should print below:

November 3rd, 2023

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
| 29 | 30 | 31 | 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 |
| 26 | 27 | 28 | 29 | 30 | | |