

```

//include library functions necessary
# include "my_functions.h"
# define INPUT_LEN 100

void print_main_menu() {
    printf("Enter D to Delete an application\n");
    printf("Enter S to Sort wins\n");
    printf("Enter M to run the hiring process\n");
    printf("Enter P to Print all applications\n");
    printf("Enter L to analyze applicant list\n");
    printf("Enter R to print same race point\n");
    printf("Enter T to terminate and write\n");
    printf("Enter X to Exit\n\n");
}

int main()
{
    int list_size;
    int clock_time = 0;
    char buffer[INPUT_LEN];
    char ch;
    struct f1player_se* head;
    struct f1player_se* pL;
    int app_to_be_gone;
    //int code;

    printf("Enter a seed to the random number generator (i.e. an integer >0)\n");
    gets(buffer);
    srand(atoi(buffer));
    head = make_init_list(clock_time);
    while (buffer[0] != 'X')
    {
        print_main_menu();
        printf(">>>> Enter a command/character ");
        fgets(buffer, INPUT_LEN, stdin);
        char switchy = toupper(buffer[0]);
        switch (switchy)
        {
            case 'D':
                print_apps(head, NULL);
                printf(">>>> Choose an Application ID to Delete\n\n");
                app_to_be_gone = (int)atoi(gets(buffer));
                pL = check_for_app(head, app_to_be_gone);
                if (pL != NULL) {
                    head = del_app(head, pL->id);
                }
                else {
                    printf("there is no application with application ID= %d\n", app_to_be_gone);
                    break;
                }
                break;

            case 'M':
                clock_time++;
                head = process(head, clock_time);
                break;
            return 1;
        }
    }
}

```

```

    case 'S':
        head = sort_wins(head);
        break;

    case 'P':
        print_apps(head, NULL);
        break;

    case 'X':
        buffer[0] = 'X';
        break;

    case 'L':
        analyze_app_list(head);
        break;

    case 'R':
        printf("Enter race point to search for\n");
        int num;
        scanf("%d", &num);
        print_samerp(head, num);
        break;

    case 'T':
        terminate_write(head);
        buffer[0] = 'X';
        break;

    default:
        printf("Enter a valid command/character\n");
}
}
}

```