

asicstyle=

1 TASK ALLOTMENT PROGRAM By SHIVANSH JAIN

2 Aim

The aim of this project is to create a task allotment system based on number of people available and number of task to be executed, using criteria of task limit per person. This project contains 10 user defined function. This project will help at organisational level to distribute and manage task based on defined inputs.

3 Following are the functions used in this project

3.1 setTaskName

This function is used to give input about Task Name .

3.2 getTaskName

This function is used to get output about Task Name.

3.3 setClubMember

This function is used to set input about Club members including task name and maximum task limit.

3.4 factorial

This function is used to find factorial.

3.5 kthPermutationUtil

This is Kth permutation utility function.

3.6 kthPermutation

This function is used to find Kth Permutation.

3.7 showallotment

This function will be called after taking all inputs to show possible allotment combination.

3.8 printClubMembers

This function is used to print club member name.

3.9 addMember

This functions will be used to add club member if required.

3.10 takeinput

This function will be used to take input about total member count, member names, task limit per person, total number of task and name of each task.

4 Code in C++ Language:

```
#include <bits/stdc++.h>
using namespace std;

class Task
{
public:
    string task_name;

    //default constructor
    Task(){}

    //constructor with 1 String argument
    Task(string s){           //constructor overloading
        task_name = s;
    }

    //function to give input about Task Name
    void setTaskName(string name){
        task_name = name;
    }

    //function to get output about Task Name
    string getTaskName(){
        return task_name;
    }
};

class ClubMember
{
public:
    string name;
    int maxTasks;

    //default constructor
    ClubMember(){}

    //constructor with 2 argument(String , integer)
    ClubMember(string n , int maxT){           //constructor overloading
        name = n;
        maxTasks = maxT;
    }
};
```

```

}

//function to set input about Club Member
void setClubMember(string s , int maxT){
    name = s;
    maxTasks = maxT;
}
};

typedef long long int ll;
class Club
{
private:
    vector<ClubMember> club_member;
    vector<Task> task_info;

// function to find factorial cd G
ll factorial(int n)
{
    ll fact = 1;
    for (int i = 1; i <= n; i++){
        fact *= i;
    }
    return fact;
}

//Kth permutation utility function
void kthPermutationUtil(vector<int> &v, vector<bool> &arr , int len , int n, ll k)
{
    if (len == 0)
    {
        return;
    }
    ll fact = factorial(len - 1);
    ll x = k / fact;
    int num;
    for (num = 0; num < n; num++)
    {
        if (arr [num] == false && ((x--) == 0))
        {
            break;
        }
    }
    v.push_back(num);
    arr [num] = true;
    kthPermutationUtil(v, arr , len - 1, n, k % fact );
}
}

```

```

//function to found Kth permutation
vector<int> kthPermutation(int n, ll k)
{
    vector<int> v;
    vector<bool> arr(n, false);
    kthPermutationUtil(v, arr, n, n, k - 1);
    return v;
}

public:
//function to show possible allotment combination
void show_allotment()
{
    int n = task_info.size();
    int mem = club_member.size();
    srand(time(0)); // seeding
    ll k = (rand() % factorial(n)) + 1;

    vector<int> ind = kthPermutation(n, k);

    for (int i = 0, j = 0; i < mem; i++)
    {
        cout << setw(10) << club_member[i].name << " : ";
        for (int l = j; l < j + club_member[i].maxTasks && l < n; l++)
        {
            cout << task_info[ind[l]].task_name << " ";
        }

        cout << "\n";
        j += club_member[i].maxTasks;
        if (j >= n){
            break;
        }
    }
}

//function to print club members name
void printClubMembers(){
    int n = club_member.size();
    for(int i=0 ; i<n ; i++){
        cout << club_member[i].name << " ";
    }
}

//function to add club member
void addMember(ClubMember mem){
    club_member.push_back(mem);
}

```

```

//function to get input about total member count, member name, task limit, total t
void take_input()
{
    int mem_count, task_count;
    cout << "Enter total no. of members: ";
    cin >> mem_count;
    cin.ignore();

    for (int i = 0; i < mem_count; i++) {
        cout << "Enter name of " << (i + 1) << " person: ";
        string str;
        getline(cin, str);
        cout << "Enter max tasks: ";
        int maxTasks; cin >> maxTasks;
        cin.ignore();
        club_member.push_back({str, maxTasks});
    }

    cout << "Enter total no. of Task to be allotted: ";
    cin >> task_count;
    cin.ignore();

    for (int i = 0; i < task_count; i++) {
        string task;
        cout << "Enter task " << (i + 1) << " name: ";
        getline(cin, task);
        task_info.push_back({task});
    }
}

int main()
{
    Club club;
    club.take_input();
    club.show_allotment();

    return 0;
}

```

5 C++ output

```
PS C:\Users\hp> cd "g:\" ; if ($?) { g++ TaskAlloter.cpp -o TaskAlloter } ; if ($?) { .\TaskAlloter
Enter total no of members: 5
Enter name of 1 person : shivansh
Enter max tasks : 2
Enter name of 2 person : tanisha
Enter max tasks : 2
Enter name of 3 person : saksham
Enter max tasks : 1
Enter name of 4 person : yash
Enter max tasks : 1
Enter name of 5 person : shubhra
Enter max tasks : 2
Enter total no of Task to be allotted : 10
Enter task 1 name : content writing
Enter task 2 name : graphic designing
Enter task 3 name : video editing
Enter task 4 name : management
Enter task 5 name : logistics
Enter task 6 name : operations
Enter task 7 name : sponsor arrangement
Enter task 8 name : HR Management
Enter task 9 name : Mentor permission
Enter task 10 name : Social Media Handles
    shivansh : content writing graphic designing
    tanisha : logistics video editing
    saksham : operations
    yash : management
    shubhra : HR Management sponsor arrangement
PS G:\> █
```

Figure 1: C++ OUTPUT

6 C++ Profiling Output:

```

hp@SJ_MINGW64 ~/s1 (master)
$ gprof a.exe gmon.out
BFD: Dwarf Error: Could not find abbrev number 108.
Flat profile:

Each sample counts as 0.01 seconds.
no time accumulated

% cumulative self total
time seconds seconds calls Ts/call Ts/call name
0.00 0.00 0.00 69 0.00 0.00 std::_Bit_iterator_base::_Bit_iterator_base(unsigned long, unsigned int)
0.00 0.00 0.00 66 0.00 0.00 std::_Bit_iterator::_Bit_iterator(unsigned long, unsigned int)
0.00 0.00 0.00 65 0.00 0.00 std::_Bit_iterator::operator() const
0.00 0.00 0.00 65 0.00 0.00 std::_Bit_reference::_Bit_reference(unsigned long, unsigned long)
0.00 0.00 0.00 65 0.00 0.00 std::_Bit_iterator::operator<std::_Bit_iterator>::operator[](unsigned int)
0.00 0.00 0.00 55 0.00 0.00 std::_Bit_reference::operator bool() const
0.00 0.00 0.00 50 0.00 0.00 Task&& std::forward<Task>(std::remove_reference<Task>::type&)
0.00 0.00 0.00 47 0.00 0.00 operator new(unsigned int, void*)
0.00 0.00 0.00 40 0.00 0.00 Task* std::__addressof<Task>(Task*)
0.00 0.00 0.00 35 0.00 0.00 Task::~Task()
0.00 0.00 0.00 30 0.00 0.00 std::move_iterator<ClubMember*>::base() const
0.00 0.00 0.00 30 0.00 0.00 int* std::__niter_base<int>::(int*)
0.00 0.00 0.00 26 0.00 0.00 ClubMember&& std::forward<ClubMember>(std::remove_reference<ClubMember>::type&)
0.00 0.00 0.00 25 0.00 0.00 Task::Task(Task&&)
0.00 0.00 0.00 25 0.00 0.00 int const& std::forward<int const&>(std::remove_reference<int const&>::type&)
0.00 0.00 0.00 25 0.00 0.00 void std::__destroy<Task>(Task*)
0.00 0.00 0.00 25 0.00 0.00 bool std::operator==<Task>(std::move_iterator<Task> const&, std::move_iterator<Task> const&)
0.00 0.00 0.00 25 0.00 0.00 std::operator!=<Task>(std::move_iterator<Task> const&, std::move_iterator<Task> const&)
0.00 0.00 0.00 24 0.00 0.00 std::vector<Task, std::allocator<ClubMember*>::operator[](unsigned int)
0.00 0.00 0.00 21 0.00 0.00 std::vector<Task, std::allocator<Task>::size() const
0.00 0.00 0.00 20 0.00 0.00 __gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task> >::base() const
0.00 0.00 0.00 20 0.00 0.00 __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> >::base() const
0.00 0.00 0.00 20 0.00 0.00 std::move_iterator<int*>::base() const
0.00 0.00 0.00 20 0.00 0.00 std::move_iterator<int>(std::move_iterator<Task>)
0.00 0.00 0.00 20 0.00 0.00 std::move_iterator<int>(std::move_iterator<Task>)
0.00 0.00 0.00 20 0.00 0.00 __dec_type<__niter_base<__ppar#1, base0>() std::__niter_base<int>::(std::move_iterator<int>)
0.00 0.00 0.00 20 0.00 0.00 int* std::__niter_base<int>::(int*)
0.00 0.00 0.00 20 0.00 0.00 std::move_iterator<Task> std::__make_move_if_noexcept<iterator<Task, std::move_iterator<Task> >::(Task*)
0.00 0.00 0.00 20 0.00 0.00 std::move_iterator<int> std::__make_move_if_noexcept<iterator<int, std::move_iterator<int> >::(int*)
0.00 0.00 0.00 19 0.00 0.00 ClubMember* std::__addressof<ClubMember>(ClubMember&)

```

```

v.vv v.vv v.vv t.v v.vv v.vv 11c 11c 11c 11c 11c 11c 11c
0.00 0.00 0.00 10 0.00 0.00 int* std::__copy_move_a<true, int*, int*, int*, int*, int*, int*>
0.00 0.00 0.00 10 0.00 0.00 Task* std::uninitialized_copy<std::move_iterator<Task>, Task>(std::move_iterator<Task>, std::move_iterator<Task>, Task)
0.00 0.00 0.00 10 0.00 0.00 int* std::uninitialized_copy<int*, int*, int*, int*, int*, int*>
0.00 0.00 0.00 10 0.00 0.00 Task* std::uninitialized_copy<std::move_iterator<Task>, Task>(std::move_iterator<Task>, std::move_iterator<Task>, Task)
) 0.00 0.00 0.00 10 0.00 0.00 int* std::__uninitialized_copy_a<std::move_iterator<int>, int*, int, int>(std::move_iterator<int>, std::move_iterator<int>, int, int)
0.00 0.00 0.00 10 0.00 0.00 Task* std::__uninitialized_copy_a<Task>::(Task*, Task, std::allocator<Task> >(Task*, Task, Task)
0.00 0.00 0.00 10 0.00 0.00 int* std::__uninitialized_move_if_noexcept_a<int, int, int, int>(std::allocator<int> >(int*, int*, int, int))
0.00 0.00 0.00 10 0.00 0.00 int* std::copy<std::move_iterator<int>, int*, int>(std::move_iterator<int>, std::move_iterator<int>, int)
0.00 0.00 0.00 10 0.00 0.00 std::remove_reference<int&&::type>::(std::move_iterator<Task>, Task)
0.00 0.00 0.00 9 0.00 0.00 std::vector<Task, std::allocator<Task> >::operator[](unsigned int)
0.00 0.00 0.00 8 0.00 0.00 std::move_iterator<Task, std::allocator<ClubMember>, std::vector<ClubMember, std::allocator<ClubMember> >::norm
0.00 0.00 0.00 8 0.00 0.00 __gnu_cxx::__normal_iterator<ClubMember, std::allocator<ClubMember> >::operator[](unsigned int)
0.00 0.00 0.00 8 0.00 0.00 std::vector<ClubMember, std::allocator<ClubMember> >::max_size() const
0.00 0.00 0.00 8 0.00 0.00 std::allocator_traits<std::allocator<ClubMember> >::max_size(std::allocator<ClubMember> const&)
0.00 0.00 0.00 8 0.00 0.00 ClubMember* std::__uninitialized_copy<false>::__uninit_copy<std::move_iterator<ClubMember>, ClubMember>
r<ClubMember*>, ClubMember*)
0.00 0.00 0.00 8 0.00 0.00 ClubMember* std::uninitialized_copy<std::move_iterator<ClubMember>, ClubMember>(std::move_iterator<ClubMember>, ClubMember*)
er*) 0.00 0.00 0.00 8 0.00 0.00 ClubMember* std::__uninitialized_copy<std::move_iterator<ClubMember>, ClubMember*, ClubMember>(std::move_iterator<ClubMember>, ClubMember*, std::allocator<ClubMember>)
0.00 0.00 0.00 8 0.00 0.00 ClubMember* std::__uninitialized_move_if_noexcept_a<ClubMember*, ClubMember*, std::allocator<ClubMember> ator<ClubMember>&)
0.00 0.00 0.00 7 0.00 0.00 std::move_iterator<ClubMember>::operator() const
0.00 0.00 0.00 7 0.00 0.00 std::move_iterator<ClubMember>::operator++()
0.00 0.00 0.00 7 0.00 0.00 void std::__Construct<ClubMember, ClubMember>(ClubMember*, ClubMember&)
0.00 0.00 0.00 6 0.00 0.00 void std::__destroy_aux<false>::__destroy<Task>(Task*, Task*)
0.00 0.00 0.00 6 0.00 0.00 void std::__destroy_aux<true>::__destroy<int>(int*, int*)
0.00 0.00 0.00 6 0.00 0.00 std::vector<Task, std::allocator<Task> >::__M_deallocate(Task*, unsigned int)
0.00 0.00 0.00 6 0.00 0.00 std::vector<Task, std::allocator<Task> >::operator[](unsigned int)
0.00 0.00 0.00 6 0.00 0.00 void std::__Destroy<Task>(Task*, Task, std::allocator<Task> &)
0.00 0.00 0.00 6 0.00 0.00 void std::__Destroy<int>(int*, int*)
0.00 0.00 0.00 6 0.00 0.00 void std::__Destroy<int>(int*, int*)
0.00 0.00 0.00 5 0.00 0.00 ClubMember* ClubMember(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >, std::allocator<ClubMember> >(ClubMember*, ClubMember&)
0.00 0.00 0.00 5 0.00 0.00 __gnu_cxx::__new_allocator<Task>::deallocate(Task*, unsigned int)
0.00 0.00 0.00 5 0.00 0.00 __gnu_cxx::__new_allocator<Task>::allocate(unsigned int, void const*)
0.00 0.00 0.00 5 0.00 0.00 __gnu_cxx::__new_allocator<Task>::allocate(int*, unsigned int)
0.00 0.00 0.00 5 0.00 0.00 __gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task> >::difference_type __gnu_cx
<Task> >::__gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task> >::const&, __gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::a

```

```

0.00 0.00 0.00 4 0.00 0.00 __gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>>::differ-
tor<ClubMember, std::allocator<ClubMember>> >(&__gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> const&, __gnu_
Member, std::allocator<ClubMember>> const&)
0.00 0.00 0.00 4 0.00 0.00 std::vector<ClubMember, std::allocator<ClubMember>>::_M_check_len(unsigned int, char const*) const
0.00 0.00 0.00 4 0.00 0.00 std::vector_base<ClubMember, std::allocator<ClubMember>>::_M_allocate(unsigned int)
0.00 0.00 0.00 4 0.00 0.00 std::allocator_traits<std::allocator<ClubMember>>::deallocate(std::allocator<ClubMember>&, ClubMember*, 
0.00 0.00 0.00 4 0.00 0.00 std::allocator_traits<std::allocator<ClubMember>>::allocate(std::allocator<ClubMember>&, unsigned int)
0.00 0.00 0.00 4 0.00 0.00 void std::vector<ClubMember, std::allocator<ClubMember>>::_M_realloc_insert<ClubMember>(__gnu_cxx::__no
td::allocator<ClubMember>> >, ClubMember&&)
0.00 0.00 0.00 4 0.00 0.00 std::vector<ClubMember, std::allocator<ClubMember>>::end()
0.00 0.00 0.00 4 0.00 0.00 std::vector<ClubMember, std::allocator<ClubMember>>::begin()
0.00 0.00 0.00 3 0.00 0.00 std::__Bit_iterator::__Bit_iterator()
0.00 0.00 0.00 3 0.00 0.00 __gnu_cxx::new_allocator<unsigned long>::__new_allocator()
0.00 0.00 0.00 2 0.00 0.00 std::vector_base<std::allocator<bool>>::_M_end_addr() const
0.00 0.00 0.00 2 0.00 0.00 std::vector_base<std::allocator<bool>>::_S_nword(unsigned int)
0.00 0.00 0.00 1 0.00 0.00 std::vector<ClubMember, std::allocator<ClubMember>>::Club::task_input()
0.00 0.00 0.00 1 0.00 0.00 Club::kthPermutation(int, long long)
0.00 0.00 0.00 1 0.00 0.00 Club::show_allotment()
0.00 0.00 0.00 1 0.00 0.00 Club::kthPermutationUtil(std::vector<int, std::allocator<int>> &, std::vector<bool, std::allocator<bool>
0.00 0.00 0.00 1 0.00 0.00 Club::Club()
0.00 0.00 0.00 1 0.00 0.00 Club::~Club()
0.00 0.00 0.00 1 0.00 0.00 __gnu_cxx::new_allocator<ClubMember>::__new_allocator()
0.00 0.00 0.00 1 0.00 0.00 __gnu_cxx::new_allocator<ClubMember>::__new_allocator()
0.00 0.00 0.00 1 0.00 0.00 __gnu_cxx::new_allocator<Task>::__new_allocator()
0.00 0.00 0.00 1 0.00 0.00 __gnu_cxx::new_allocator<Task>::__new_allocator()
0.00 0.00 0.00 1 0.00 0.00 __gnu_cxx::new_allocator<bool>::__new_allocator()
0.00 0.00 0.00 1 0.00 0.00 __gnu_cxx::new_allocator<bool>::__new_allocator()
0.00 0.00 0.00 1 0.00 0.00 __gnu_cxx::new_allocator<int>::__new_allocator()
0.00 0.00 0.00 1 0.00 0.00 __gnu_cxx::new_allocator<int>::__new_allocator()
0.00 0.00 0.00 1 0.00 0.00 __gnu_cxx::new_allocator<unsigned long>::deallocate(unsigned long*, unsigned int)
0.00 0.00 0.00 1 0.00 0.00 __gnu_cxx::new_allocator<unsigned long>::allocate(unsigned int, void const*)
0.00 0.00 0.00 1 0.00 0.00 __gnu_cxx::new_allocator<unsigned long>::__new_allocator(__gnu_cxx::__new_allocator<unsigned long> const)
0.00 0.00 0.00 1 0.00 0.00 __gnu_cxx::new_allocator<unsigned long>::max_size() const
0.00 0.00 0.00 1 0.00 0.00 std::__Bit_iterator::operator+(int) const
0.00 0.00 0.00 1 0.00 0.00 std::allocator<ClubMember>::allocator()
0.00 0.00 0.00 1 0.00 0.00 std::allocator<ClubMember>::~allocator()
0.00 0.00 0.00 1 0.00 0.00 std::allocator<Task>::allocator()
0.00 0.00 0.00 1 0.00 0.00 std::allocator<Task>::~allocator()
0.00 0.00 0.00 1 0.00 0.00 std::allocator<bool>::allocator()
0.00 0.00 0.00 1 0.00 0.00 std::allocator<bool>::~allocator()
0.00 0.00 0.00 1 0.00 0.00 std::allocator<bool>::__allocat

```

```

0.00 0.00 10/47 void __gnu_cxx::new_allocator<Task>::construct<Task, Task>(Task*, Task&&) [55]
0.00 0.00 10/47 void __gnu_cxx::new_allocator<int>::construct<int, int const&>(int*, int const&&) [56]
0.00 0.00 15/47 void std::__construct<Task, Task>(Task*, Task&&) [45]
[10] 0.0 0.0 0.0 47 operator new(unsigned int, void*) [10]
-----
0.00 0.00 15/40 Task* std::__uninitialized_copy<false>::__uninit_copy<std::move_iterator<Task*>, Task>(std::move_iterator<T
0.00 0.00 25/40 void std::__Destroy_aux<false>::__destroy<Task*>(Task*, Task*) [97]
[11] 0.0 0.0 0.0 40 Task* std::__addressof<Task>(Task&) [11]
-----
0.00 0.00 10/35 _ful15__Zst3cin [303]
0.00 0.00 25/35 void std::__destroy<Task>(Task*) [18]
[12] 0.0 0.0 0.0 35 Task::~Task() [12]
-----
0.00 0.00 30/30 bool std::operator==<ClubMember>(std::move_iterator<ClubMember> const&, std::move_iterator<ClubMember> co
[13] 0.0 0.0 0.0 30 std::move_iterator<ClubMember>::base() const [13]
-----
0.00 0.00 30/30 int* std::__copy_move_a2<true, int*, int*>(int*, int*, int*) [75]
[14] 0.0 0.0 0.0 30 int* std::__niter_base<int*>(int*) [14]
-----
0.00 0.00 4/26 void std::vector<ClubMember, std::allocator<ClubMember>>::_M_realloc_insert<ClubMember>(__gnu_cxx::__normal_
locator<ClubMember> >, ClubMember&&) [143]
0.00 0.00 5/26 void __gnu_cxx::new_allocator<ClubMember>::__construct<ClubMember, ClubMember>(ClubMember*, ClubMember&&) [106]
0.00 0.00 5/26 void std::allocator_traits<std::allocator<ClubMember>>::construct<ClubMember, ClubMember>(std::allocator<Cl
0.00 0.00 5/26 void std::vector<ClubMember, std::allocator<ClubMember>>::emplace_back<ClubMember>(ClubMember&&) [124]
0.00 0.00 7/26 void std::vector<ClubMember, std::allocator<ClubMember>>::push_back(ClubMember&&) [96]
[15] 0.0 0.0 0.0 26 ClubMember&& std::forward<ClubMember>(std::remove_reference<ClubMember>::type&) [15]
-----
0.00 0.00 10/25 void __gnu_cxx::new_allocator<Task>::construct<Task, Task>(Task*, Task&&) [55]
0.00 0.00 15/25 void std::__construct<Task, Task>(Task*, Task&&) [45]
[16] 0.0 0.0 0.0 25 Task::Task(Task&&) [16]
-----
0.00 0.00 5/25 void std::vector<int, std::allocator<int>>::_M_realloc_insert<int const>(__gnu_cxx::__normal_iterator<int>
@ [129]
0.00 0.00 10/25 void __gnu_cxx::new_allocator<int>::__construct<int, int const&>(int*, int const&) [56]
0.00 0.00 10/25 void std::allocator_traits<std::allocator<int>>::construct<int, int const&>(std::allocator<int> const&, int*
[17] 0.0 0.0 0.0 25 int const& std::forward<int const>(std::remove_reference<int const&>::type&) [17]
-----
0.00 0.00 25/25 void std::__Destroy_aux<false>::__destroy<Task*>(Task*, Task*) [97]
[18] 0.0 0.0 0.0 25 void std::__destroy<Task>(Task*) [18]
0.00 0.00 25/35 Task::~Task() [12]
-----
0.00 0.00 25/25 bool std::operator!=<Task*>(std::move_iterator<Task*> const&, std::move_iterator<Task*> const&) [20]
[19] 0.0 0.0 0.0 25 bool std::operator==<Task*>(std::move_iterator<Task*> const&, std::move_iterator<Task*> const&) [19]
0.00 0.00 50/50 std::move_iterator<Task*>::base() const [8]

```

```

0.00 0.00 25/25      bool std::operator==(Task&)(std::move_iterator<Task> const&, std::move_iterator<Task> const&) [19]
-----
0.00 0.00 5/24      _fu2__zst4cout [305]
0.00 0.00 24/24      _fu0__zst4cout [297]
[21] 0.0 0.00 0.00 24 std::vector<ClubMember, std::allocator<ClubMember>>::operator[](unsigned int) [21]
-----
0.00 0.00 1/21      Club::show_allotment() [153]
0.00 0.00 20/21      std::vector<Task, std::allocator<Task>>::_M_check_len(unsigned int, char const*) const [113]
[22] 0.0 0.00 0.00 21 std::vector<Task, std::allocator<Task>>::size() const [22]
-----
0.00 0.00 10/20      __gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>>::difference_type __gnu_cxx::operator>>>(<__gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>>> const&, __gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const&)
0.00 0.00 10/20      void std::vector<Task, std::allocator<Task>>::_M_realloc_insert<Task>(<__gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const&, __gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const&)
[26] 0.0 0.00 0.00 20 __gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>>::base() const [23]
-----
0.00 0.00 10/20      __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int>>::difference_type __gnu_cxx::operator>>>(<__gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int>> const&, __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int>> const&)
0.00 0.00 10/20      void std::vector<int, std::allocator<int>>::_M_realloc_insert<int>(<__gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int>> const&, __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int>> const&)
[24] 0.0 0.00 0.00 20 __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int>>::base() const [24]
-----
0.00 0.00 20/20      decltype (<__miter_base((&param#1).base()) std::__miter_base<int>(std::move_iterator<int>)) [29]
[25] 0.0 0.00 0.00 20 std::move_iterator<int>::base() const [25]
-----
0.00 0.00 20/20      std::vector<int, std::allocator<int>>::_M_check_len(unsigned int, char const*) const [114]
[26] 0.0 0.00 0.00 20 std::vector<int, std::allocator<int>>::size() const [26]
-----
0.00 0.00 20/20      std::move_iterator<Task> std::__make_move_if_noexcept_iterator<Task, std::move_iterator<Task>>(<Task>) [31]
[27] 0.0 0.00 0.00 20 std::move_iterator<Task>::move_iterator(Task*) [27]
-----
0.00 0.00 20/20      std::move_iterator<int> std::__make_move_if_noexcept_iterator<int, std::move_iterator<int>>(<int>) [32]
[28] 0.0 0.00 0.00 20 std::move_iterator<int>::move_iterator(int*) [28]
-----
0.00 0.00 20/20      int* std::copy<std::move_iterator<int>, int>(<std::move_iterator<int>, std::move_iterator<int>, int) [82]
[29] 0.0 0.00 0.00 20 decltype (<__miter_base((&param#1).base()) std::__miter_base<int>(std::move_iterator<int>)) [29]
0.00 0.00 20/20      std::move_iterator<int>::base() const [25]
0.00 0.00 20/20      int* std::__miter_base<int>(int*) [30]
-----
0.00 0.00 20/20      decltype (<__miter_base((&param#1).base()) std::__miter_base<int>(std::move_iterator<int>)) [29]
[30] 0.0 0.00 0.00 20 int* std::__miter_base<int>(int*) [30]
-----
0.00 0.00 20/20      Task* std::__uninitialized_move_if_noexcept_a<Task*, Task*, std::allocator<Task>>(<Task*, Task*, Task*, std::move_iterator<Task>)
[31] 0.0 0.00 0.00 20 std::move_iterator<Task> std::__make_move_if_noexcept_iterator<Task, std::move_iterator<Task>>(<Task>) [31]

```

```

0.00 0.00 5/5 void std::_Destroy_aux<false>::__destroy<ClubMember>(ClubMember*, ClubMember*) [115]
-----
0.00 0.00 1/5 std::vector<ClubMember, std::allocator<ClubMember>>::vector() [208]
0.00 0.00 4/5 void std::vector<ClubMember, std::allocator<ClubMember>>::__M_realloc_insert<ClubMember>(__gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> >, ClubMember&&) [143]
[135] 0.0 0.0 0.0 5 void std::destroy<ClubMember>(ClubMember*, ClubMember*, std::allocator<ClubMember>&) [135]
0.0 0.0 0.0 5/5 void std::__Destroy<ClubMember>(ClubMember*, ClubMember*) [134]
-----
0.00 0.00 4/4 std::allocator_traits<std::allocator<ClubMember>>::deallocate(std::allocator<ClubMember>&, ClubMember*, unsigned int) [141]
[136] 0.0 0.0 0.0 4 __gnu_cxx::new_allocator<ClubMember>::deallocate(ClubMember*, unsigned int) [136]
0.00 0.00 4/4 std::allocator_traits<std::allocator<ClubMember>>::allocate(std::allocator<ClubMember>&, unsigned int) [142]
[137] 0.0 0.0 0.0 4 __gnu_cxx::new_allocator<ClubMember>::allocate(unsigned int, void const*) [137]
0.00 0.00 4/12 __gnu_cxx::new_allocator<ClubMember>::max_size() const [51]
-----
0.00 0.00 4/4 void std::vector<ClubMember, std::allocator<ClubMember>>::__M_realloc_insert<ClubMember>(__gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> >, ClubMember&&) [143]
[138] 0.0 0.0 0.0 4 __gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>>::__difference_type __gnu_cxx::operator-<ClubMember>, std::vector<ClubMember, std::allocator<ClubMember>> >> const&, __gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> >> const& [138]
0.0 0.0 0.0 4/16 __gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>>::__base() const [36]
-----
0.00 0.00 4/4 void std::vector<ClubMember, std::allocator<ClubMember>>::__M_realloc_insert<ClubMember>(__gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> >, ClubMember&&) [143]
[139] 0.0 0.0 0.0 4 std::vector<ClubMember, std::allocator<ClubMember>>::__M_check_len(unsigned int, char const*) const [139]
0.00 0.00 16/17 std::vector<ClubMember, std::allocator<ClubMember>>::__size() const [35]
0.00 0.00 8/8 std::vector<ClubMember, std::allocator<ClubMember>>::__max_size() const [88]
0.00 0.00 4/34 unsigned int const& std::max<unsigned int>(unsigned int const&, unsigned int const&) [48]
-----
0.00 0.00 4/4 void std::vector<ClubMember, std::allocator<ClubMember>>::__M_realloc_insert<ClubMember>(__gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> >, ClubMember&&) [143]
[140] 0.0 0.0 0.0 4 std::vector_base<ClubMember, std::allocator<ClubMember>>::__M_allocate(unsigned int) [140]
0.00 0.00 4/4 std::allocator_traits<std::allocator<ClubMember>>::allocate(std::allocator<ClubMember>&, unsigned int) [142]
-----
0.00 0.00 4/4 std::vector_base<ClubMember, std::allocator<ClubMember>>::__M_deallocate(ClubMember*, unsigned int) [116]
[141] 0.0 0.0 0.0 4 std::allocator_traits<std::allocator<ClubMember>>::deallocate(std::allocator<ClubMember>&, ClubMember*, unsigned int) [141]
0.00 0.00 4/4 __gnu_cxx::new_allocator<ClubMember>::deallocate(ClubMember*, unsigned int) [136]
-----
0.00 0.00 4/4 std::vector_base<ClubMember, std::allocator<ClubMember>>::__M_allocate(unsigned int) [140]
[142] 0.0 0.0 0.0 4 std::allocator_traits<std::allocator<ClubMember>>::allocate(std::allocator<ClubMember>&, unsigned int) [142]
0.00 0.00 4/4 __gnu_cxx::new_allocator<ClubMember>::allocate(unsigned int, void const*) [137]
-----
0.00 0.00 4/4 void std::vector<ClubMember, std::allocator<ClubMember>>::__emplace_back<ClubMember>(ClubMember&&) [124]
[143] 0.0 0.0 0.0 4 std::vector<ClubMember, std::allocator<ClubMember>>::__M_realloc_insert<ClubMember>(__gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> >, ClubMember&&) [143]
0.00 0.00 12/13 std::vector_base<ClubMember, std::allocator<ClubMember>>::__M_get_tp_allocator() [49]
0.00 0.00 8/16 __gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>>::__base() const [36]
0.00 0.00 8/8 ClubMember* std::__uninitialized_move_if_noexcept<ClubMember*, ClubMember*, std::allocator<ClubMember>> [93]
0.00 0.00 4/4 std::vector<ClubMember, std::allocator<ClubMember>>::__M_check_len(unsigned int, char const*) const [139]
0.00 0.00 4/4 std::vector<ClubMember, std::allocator<ClubMember>>::__begin() [145]
0.00 0.00 4/4 __gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>>::__difference_type __gnu_cxx::operator-<ClubMember>, std::vector<ClubMember, std::allocator<ClubMember>> >> const&, __gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> >> const& [138]
0.00 0.00 4/26 std::vector_base<ClubMember, std::allocator<ClubMember>>::__remove_reference<ClubMember>::type@0 [15]
0.00 0.00 4/5 void std::allocator_traits<std::allocator<ClubMember>>::__construct<ClubMember, ClubMember>(std::allocator<ClubMember>&, ClubMember*, ClubMember&&) [119]
0.00 0.00 4/5 void std::destroy<ClubMember>(ClubMember*, ClubMember*, std::allocator<ClubMember>&) [135]
0.00 0.00 4/5 std::__vector_base<ClubMember, std::allocator<ClubMember>>::__M_deallocate(ClubMember*, unsigned int) [116]
-----
0.00 0.00 4/4 void std::vector<ClubMember, std::allocator<ClubMember>>::__emplace_back<ClubMember>(ClubMember&&) [124]
[144] 0.0 0.0 0.0 4 std::vector<ClubMember, std::allocator<ClubMember>>::__end() [144]
0.00 0.00 4/8 __gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> >::__normal_iterator(ClubMember* const&) [86]
0.00 0.00 4/4 void std::vector<ClubMember, std::allocator<ClubMember>>::__M_realloc_insert<ClubMember>(__gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> >, ClubMember&&) [143]

```

[214]	0.0	0.00	0.00	1/1	club::kthPermutation(int, long long) [152] std::vector<bool, std::allocator<bool> >::~vector() [214] std::_Bvector_base<std::allocator<bool> >::~_Bvector_base() [203]
[215]	0.0	0.00	0.00	1/1	club::kthPermutation(int, long long) [152] std::vector<int, std::allocator<int> >::vector() [215] std::_Vector_base<int, std::allocator<int> >::_Vector_base() [193]
[216]	0.0	0.00	0.00	1/1	_fu2__Zst4cout [305] std::vector<int, std::allocator<int> >::~vector() [216] std::_Vector_base<int, std::allocator<int> >::_Vector_base() [38] void std::_Destroy<int*, int>(int*, int*, std::allocator<int>&) [104] std::_Vector_base<int, std::allocator<int> >::~_Vector_base() [194]

This table describes the call tree of the program, and was sorted by the total amount of time spent in each function and its children.

Each entry in this table consists of several lines. The line with the index number at the left hand margin lists the current function. The lines above it list the functions that called this function, and the lines below it list the functions this one called.

This line lists:

index	A unique number given to each element of the table. Index numbers are sorted numerically. The index number is printed next to every function name so it is easier to look up where the function is in the table.
% time	This is the percentage of the 'total' time that was spent in this function and its children. Note that due to different viewpoints, functions excluded by options, etc, these numbers will NOT add up to 100%.
self	This is the total amount of time spent in this function.
children	This is the total amount of time propagated into this function by its children.
called	This is the number of times the function was called. If the function called itself recursively, the number only includes non-recursive calls, and is followed by a '+' and the number of recursive calls.
name	The name of the current function. The index number is printed after it. If the function is a member of a cycle, the cycle number is printed between the function's name and the index number.

For the function's parents, the fields have the following meanings:

self	This is the amount of time that was propagated directly from the function into this parent.
children	This is the amount of time that was propagated from the function's children into this parent.
called	This is the number of times this parent called the function '/' the total number of times the function was called. Recursive calls to the function are not included in the number after the '/'.
name	This is the name of the parent. The parent's index number is printed after it. If the parent is a

the name and the index number.

If the parents of the function cannot be determined, the word `<spontaneous>` is printed in the 'name' field, and all the other fields are blank.

For the function's children, the fields have the following meanings:

<code>self</code>	This is the amount of time that was propagated directly from the child into the function.
<code>children</code>	This is the amount of time that was propagated from the child's children to the function.
<code>called</code>	This is the number of times the function called this child '/' the total number of times the child was called. Recursive calls by the child are not listed in the number after the '/'.
<code>name</code>	This is the name of the child. The child's index number is printed after it. If the child is a member of a cycle, the cycle number is printed between the name and the index number.

If there are any cycles (circles) in the call graph, there is an entry for the cycle-as-a-whole. This entry shows who called the cycle (as parents) and the members of the cycle (as children.) The '+' recursive calls entry shows the number of function calls that were internal to the cycle, and the calls entry for each member shows, for that member, how many times it was called from other members of the cycle.

Copyright (C) 2012-2018 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification, are permitted in any medium without royalty provided the copyright notice and this notice are preserved.

Index by function name

```
[105] ClubMember::ClubMember(std::cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >, int) [174] std::allocator<Task>::~allocator() [124] void std::vector<ClubMember, std::allocator<ClubMember>>::emplace_back(ClubMember&&) [50] ClubMember(ClubMember&&) [175] std::allocator<bool>::allocator() [143] void std::vector<ClubMember, std::allocator<ClubMember>>::M_realloc_insert(<ClubMember>){_gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>>::begin(), _gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>>::end()}_ [34] ClubMember::~ClubMember() [176] std::allocator<bool>::allocator() [144] std::vector<ClubMember, std::allocator<ClubMember>>::end() [151] Club::take_input() [177] std::allocator<int>::allocator() [145] std::vector<ClubMember, std::allocator<ClubMember>>::begin() [152] Club::kthpermutation(int, long long) [178] std::allocator<int>::allocator() [125] std::vector<ClubMember, std::allocator<ClubMember>>::push_back(ClubMember&&) [153] Club::show_allotment() [179] std::allocator<unsigned long>::allocator<bool>(std::allocator<bool> const&) [207] std::vector<ClubMember, std::allocator<ClubMember>>::vector() [154] Club::kthpermutationutil(std::vector<int, std::allocator<int>>&, std::vector<bool, std::allocator<bool> &, int, int, long long) [180] std::allocator<unsigned long>::allocator(std::allocator<unsigned long> const&) [208] std::vector<ClubMember, std::allocator<ClubMember>>::vector() [155] Club::factorial(int) [181] std::allocator<unsigned long>::allocator() [21] std::vector<ClubMember, std::allocator<ClubMember>>::operator[](unsigned int) [156] Club::Club() [182] std::allocator<unsigned long>::allocator() [71] void std::vector<Task, std::allocator<Task>>::emplace_back<Task>(Task&&) [157] Club::~Club() [183] im< std::copy_move<true, true, std::random_access_iterator_tag>;_copy_m<int>(int const*, int const*, int*) [126] void std::vector<Task, std::allocator<Task>>::insert<Task>(_gnu_cxx::__normal_iterator<Task>, std::vector<Task, std::allocator<Task>>::begin(), Task&&) [158] Club::Club(std::cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >) [115] void std::Destroy_aux<false>::__destroy<ClubMember>(ClubMember*, ClubMember*) [127] std::vector<Task, std::allocator<Task>>::end() [16] Task::Task(Task&&) [97] void std::Destroy_aux<false>::__destroy<Task>(Task*, Task*) [128] std::vector<Task, std::allocator<Task>>::begin() [12] Task::~Task() [98] void std::Destroy_aux<true>::__destroy<int>(int*, int*) [72] std::vector<Task, std::allocator<Task>>::push_back(Task&&) [136] _gnu_cxx::new_allocator<ClubMember>::deallocate(ClubMember*, unsigned int) [140] std::vector<ClubMember, std::allocator<ClubMember>>::M_allocate(unsigned int) [209] std::vector<Task, std::allocator<Task>>::vector() [137] _gnu_cxx::new_allocator<ClubMember>::allocate(unsigned int, void const*) [183] std::vector<ClubMember, std::allocator<ClubMember>>::__vector_impl::__Vector_impl() [210] std::vector<Task,
```

```

[58] __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int>>::__normal_iterator(int* const&) [196] std::__bvector_base<std::allocator<bool>>::__M_allocate(unsigned int) [75] int*
y_move_a2<true, int*, int*>(int*, int*, int*)
[138] __gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>>::difference_type __gnu_cxx::operator-<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>>::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> const&, __gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> > [197] std::__bvector_base<std::allocator<bool>>::__bvector_Impl::std::__bvector_base<std::allocator<unsigned long>, const&> [91] ClubMember std::uninitialized_copy<std::move_iterator<ClubMember>, ClubMember>::operator=(__gnu_cxx::__normal_iterator<ClubMember*, std::move_iterator<ClubMember> const&)
[112] __gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>>::operator-(__gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const&, __gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const) [198] std::__bvector_base<std::allocator<bool>>::__bvector_Impl::std::__bvector_Impl::operator-(<__gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const&, __gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const) [112] Task* std::uninitialized_copy<std::move_iterator<Task>, Task>(std::move_iterator<Task>, Task*) [199] std::__bvector_base<std::allocator<bool>>::__bvector_Impl::operator-(<__gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const&, __gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const) [199] std::__bvector_base<std::allocator<bool>>::~__bvector_Impl() [77] int* __t initialized_copy<std::move_iterator<int>, std::move_iterator<int> const [200] std::__bvector_base<std::allocator<bool>>::__bvector_Impl::operator-(<__gnu_cxx::__normal_iterator<int>, __gnu_cxx::__normal_iterator<int> const) [51] __gnu_cxx::new_allocator<ClubMember>::max_size() const [200] std::__bvector_base<std::allocator<bool>>::__bvector_Impl::operator-(<__gnu_cxx::__normal_iterator<int>, __gnu_cxx::__normal_iterator<int> const) [201] std::__bvector_base<std::allocator<bool>>::__bvector_Impl::operator-(<__gnu_cxx::__normal_iterator<int>, __gnu_cxx::__normal_iterator<int> const) [78] Task* std::uninitialized_copy<_A<std::move_iterator<int>, int>, int>(std::move_iterator<int>, _A<std::move_iterator<int>, int> const) [41] __gnu_cxx::new_allocator<Task>::max_size() const [201] std::__bvector_base<std::allocator<bool>>::__bvector_Impl::operator-(<__gnu_cxx::__normal_iterator<int>, __gnu_cxx::__normal_iterator<int> const) [79] int* std::uninitialized_copy<_A<std::move_iterator<int>, int>, int>(std::move_iterator<int>, _A<std::move_iterator<int>, int> const) [42] __gnu_cxx::new_allocator<int>::max_size() const [190] std::__bvector_base<std::allocator<bool>>::__bvector_Impl::operator-(<__gnu_cxx::__normal_iterator<int>, __gnu_cxx::__normal_iterator<int> const) [79] int* std::uninitialized_copy<_A<std::move_iterator<int>, int>, int>(std::move_iterator<int>, _A<std::move_iterator<int>, int> const) [112] __gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>>::operator-(<__gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> const&, __gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> const) [40] std::move_iterator<ClubMember>::operator-(<__gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> const&, __gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> const) [36] __gnu_cxx::new_allocator<ClubMember>::operator-(<__gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> const&, __gnu_cxx::__normal_iterator<ClubMember*, std::vector<ClubMember, std::allocator<ClubMember>> const) [203] std::__bvector_base<std::allocator<bool>>::__bvector_base() [31] std::move_if_noexcept<__gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const, __gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const> [23] __gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>>::base() const [39] std::move_iterator<ClubMember>::operator-(<__gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const) [32] std::move_iterator<int>::operator-(<__gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const) [24] __gnu_cxx::__normal_iterator<int>::operator-(<__gnu_cxx::__normal_iterator<int>, __gnu_cxx::__normal_iterator<int> const) [95] std::move_iterator<ClubMember>::operator++() [93] ClubMember std::uninitialized_move_if_noexcept<ClubMember, ClubMember>::operator-(<__gnu_cxx::__normal_iterator<ClubMember*, ClubMember*, std::allocator<ClubMember>> const&, __gnu_cxx::__normal_iterator<ClubMember*, ClubMember*, std::allocator<ClubMember>> const) [87] std::__bvector_base<ClubMember, std::allocator<ClubMember>>::M_get_Tp_allocator() const [27] std::move_iterator<Task>::operator-(<__gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const) [80] Task* std::uninitialized_move_if_noexcept<__gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const, __gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const> [59] std::__bvector_base<Task, std::allocator<Task>>::operator-(<__gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const) [44] std::move_iterator<Task>::operator++() [81] int* std::uninitialized_move_if_noexcept<__gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const, __gnu_cxx::__normal_iterator<Task*, std::vector<Task, std::allocator<Task>> const> [46] std::__bvector_base<int, std::allocator<int>>::operator-(<__gnu_cxx::__normal_iterator<int>, __gnu_cxx::__normal_iterator<int> const) [28] std::move_iterator<int>::operator-(<__gnu_cxx::__normal_iterator<int>, __gnu_cxx::__normal_iterator<int> const) [4] std::__bit_iterator::operator-() const [5] std::__bit_reference::operator-(<__bit_reference<unsigned long, unsigned long> const&, __bit_reference<unsigned long, unsigned long> const) [82] int* std::copy<std::move_iterator<int>, int>(std::move_iterator<int>, int) [170] std::__bit_iterator::operator-(int) const [64] std::__bit_reference::operator=(bool) [132] std::remove_reference<ClubMember&>::type&::move<ClubMember>(<ClubMember>) [149] std::__bvector_base<std::allocator<bool>>::__bvector_Impl::M_end_addr() const [141] std::allocator_traits<std::allocator<ClubMember>>::deallocate(std::allocator<ClubMember>, ClubMember, unsigned int) [83] std::remove_reference<Task&>::type&::move<Task>(<Task>) [13] std::move<ClubMember>::base() const [142] std::allocator_traits<std::allocator<ClubMember>>::allocate(std::allocator<ClubMember>, unsigned int) [133] std::set<int>::operator-(<__gnu_cxx::__normal_iterator<ClubMember*, ClubMember*, std::allocator<ClubMember>> const) [94] std::move<ClubMember>::operator-() const [89] std::allocator_traits<std::allocator<ClubMember>>::max_size(std::allocator<ClubMember>, const) [15] ClubMember& std::forward<ClubMember>::operator-(<ClubMember> const) [88] std::__bit_iterator::operator-(<__bit_iterator<unsigned int, unsigned int> const) [120] std::__bit_iterator::operator-(<__bit_iterator<unsigned int, unsigned int> const) [17] int const& std::forward<int>::operator-(<__reference<int> const) [25] std::__bit_iterator::operator-(<__bit_iterator<unsigned int, unsigned int> const) [121] std::__bit_iterator::operator-(<__bit_iterator<unsigned int, unsigned int> const) [52] void std::destroy<ClubMember>(<ClubMember>) [7] std::__bit_reference::operator=(bool) [65] std::__bit_iterator::operator-(<__bit_iterator<Task*, Task*, std::allocator<Task>> const) [18] void std::destroy<Task>(<Task>) [139] std::__vector<ClubMember, std::allocator<ClubMember>>::operator-(<__normal_iterator<ClubMember*, ClubMember*, std::allocator<ClubMember>> const) [66] void std::allocator_traits<std::allocator<Task>>::construct<Task>(std::allocator<Task>, Task&) [35] std::__vector<ClubMember, std::allocator<ClubMember>>::operator-(<__normal_iterator<ClubMember*, ClubMember*, std::allocator<ClubMember>> const) [134] void std::destroy<ClubMember>(<ClubMember>) [88] std::__bit_iterator::operator-(<__bit_iterator<ClubMember*, ClubMember*, std::allocator<ClubMember>> const) [123] std::__bit_iterator::operator-(<__bit_iterator<ClubMember*, ClubMember*, std::allocator<ClubMember>> const) [101] void std::destroy<_, Task>(<Task>) [113] std::__vector<Task, std::allocator<Task>>::operator-(<__normal_iterator<Task*, Task*, std::allocator<Task>> const) [67] std::__vector<Task, std::allocator<Task>>::operator-(<__normal_iterator<Task*, Task*, std::allocator<Task>> const) [102] void std::destroy<_, Task>(<Task>) [122] std::__vector<Task, std::allocator<Task>>::size() const [68] void std::allocator_traits<std::allocator<Task>>::construct<int, int const>(std::allocator<Task>, int, int const) [103] void std::destroy<_, Task>(<Task>) [61] std::__vector<Task, std::allocator<Task>>::max_size() const [204] std::__vector<Task, std::allocator<Task>>::operator-(<__normal_iterator<Task*, Task*, std::allocator<Task>> const) [114] std::__vector<Task, std::allocator<Task>>::operator-(<__normal_iterator<Task*, Task*, std::allocator<Task>> const) [205] std::__vector<Task, std::allocator<Task>>::operator-(<__normal_iterator<Task*, Task*, std::allocator<Task>> const) [26] std::__vector<Task, std::allocator<Task>>::size() const [206] std::__bit_iterator::base::__M_incr(int) [19] bool std::operator==<Task>(std::move_iterator<Task>, Task*) const& [62] std::__vector<Task, std::allocator<Task>>::operator-(<__normal_iterator<Task*, Task*, std::allocator<Task>> const) [2] std::__bit_iterator::base::__M_incr(int) const [47] bool std::operator!=<ClubMember>(std::move_iterator<ClubMember>, ClubMember) >& const& std::move<ClubMember>(<ClubMember> const) [171] std::allocator<ClubMember>::operator-(<__normal_iterator<ClubMember*, ClubMember*, std::allocator<ClubMember>> const) [90] ClubMember std::uninitialized_copy<false>::operator-(<__normal_iterator<ClubMember>, ClubMember>) (std::move<ClubMember>, ClubMember, const) [20] bool std::operator!=<Task>(std::move<Task>, Task) const
```

```

L4444 -__gnu_cxx::new_allocator<Task> > const&, __gnu_cxx::__normal_iterator<Task>, std::vector<Task, std::allocator<Task>> >::operator=(const __gnu_cxx::new_allocator<Task>, std::vector<Task, std::allocator<Task>> > const&) [198] std::::_Bvector_base<std::allocator<bool>> ::~_Bvector
._impl() [76] Task* std::uninitialized_copy<std::move_iterator<Task*>, Task*>(std::move_iterator<Task*>, std::move_iterator<Task*>, std::move_iterator<Task*>, Task*)
[112] __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int>> >::difference_type __gnu_cxx::operator-<int*, std::vector<int, std::allocator<int>> > const&) [198] std::::_Bvector_base<std::allocator<bool>> ::~_Bvector
_base<std::allocator<bool>> ::M_deallocate() [77]
[112] __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int>> >::operator-<int*, std::vector<int, std::allocator<int>> > const&) [199] std::::_Bvector_base<std::allocator<bool>> ::~_Bvector
_base<std::allocator<bool>> ::M_deallocate() [77]
[51] __gnu_cxx::new_allocator<ClubMember>::max_size() const [200] std::::_Bvector_base<std::allocator<bool>> ::~_Bvector_impl_data::M_reset() [92] ClubMember* std::uninitialized_copy_a<std::move
Iterator<int*, std::allocator<int*, int>, std::move_iterator<int*, std::move_iterator<int*, int>>(&__gnu_cxx::operator-<int*, std::vector<int, std::allocator<int>> > const&)
[41] __gnu_cxx::new_allocator<Task>::max_size() const [201] std::::_Bvector_base<std::allocator<bool>> ::~_Bvector_impl_data::Bvector_impl_data() [78] Task* std::uninitialized_copy<std::move
Iterator<int*, std::allocator<int*, int>, std::move_iterator<int*, std::move_iterator<int*, int>>(&__gnu_cxx::operator-<int*, std::vector<int, std::allocator<int>> > const&)
[42] __gnu_cxx::new_allocator<int>::max_size() const [150] std::::_Bvector_base<std::allocator<bool>> ::S_nword(unsigned int) [79] int* std::uninitialized_copy<std::move_iterator<int*, int*
Iterator<int*, std::allocator<int*, int>, std::move_iterator<int*, std::move_iterator<int*, int>>(&__gnu_cxx::operator-<int*, std::vector<int, std::allocator<int>> > const&)
[169] __gnu_cxx::new_allocator<unsigned long>::max_size() const [202] std::::_Bvector_base<std::allocator<bool>> ::~_Bvector_base<std::allocator<bool>> const) [40] std::move_iterator<ClubMember>
::if_noexcept<Iterator<ClubMember, std::move_iterator<ClubMember*> >(&ClubMember*)
[36] __gnu_cxx::__normal_iterator<ClubMember> std::vector<ClubMember, std::allocator<ClubMember>> ::base() const [203] std::::_Bvector_base<std::allocator<bool>> ::~_Bvector_base() [31] std::make
_move_if_noexcept<Iterator<ClubMember, std::move_iterator<ClubMember*> >::make_move_if_noexcept<Iterator<ClubMember, std::move_iterator<ClubMember*> >(&Task*)
[23] __gnu_cxx::__normal_iterator<Task*> std::vector<Task, std::allocator<Task>> ::base() const [39] std::move_iterator<ClubMember>::move_iterator(ClubMember*) [32] std::move_iterator<int*
Iterator<int*, std::allocator<int*, int>, std::move_iterator<int*, std::move_iterator<int*, int>>(&__gnu_cxx::operator-<int*, std::vector<int, std::allocator<int>> > const&)
[24] __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int>> >::operator++() [93] ClubMember* std::uninitialized_move_if_noe
xcept<Iterator<ClubMember, std::allocator<ClubMember*> >(&ClubMember, ClubMember*, ClubMember*) const [15] ClubMember* std::allocator<ClubMember*>::M_get_tp_allocator() const [27] std::move_iterator<Task*>::move_iterator(Task*) [80] Task* std::uninitialized_move_if_noexcept<Iterator<ClubMember, std::allocator<ClubMember*> >(&Task*)
[59] std::Vector_base<Task, std::allocator<Task>> ::M_get_tp_allocator() const [44] std::move_iterator<Task*>::operator++() [81] int* std::uninitialized_move_if_noexcept_a<int*, int*, std::move
Iterator<int*, std::allocator<int>, std::move_iterator<int*>(&__gnu_cxx::operator-<int*, std::vector<int, std::allocator<int>> > const&)
[60] std::Vector_base<int, std::allocator<int>> ::M_getTp_allocator() const [28] std::move_iterator<int*>::move_iterator(int*) [48] unsigned int const& std::max<unsigned int>(unsigned int, co
nt const&)
[4] std::Bit_iterator::operator() const [5] std::Bit_reference::Bit_reference(unsigned long, unsigned long) [82] int* std::copy<std::move_iterator<int*, int*>, std::move_iterator<int*, int*>(&__gnu_cxx::operator-<int*, int*)
[170] std::Bit_iterator::operator+(int) const [64] std::Bit_reference::operator<bool> [132] std::remove_reference<ClubMember>::type&& std::move<ClubMember>(&ClubMember)
[149] std::::_Bvector_base<std::allocator<bool>> ::M_end_addr() const [141] std::allocator_traits<std::allocator<ClubMember*> >::deallocate(std::allocator<ClubMember*>, ClubMember*, &ClubMember*)
[83] std::remove_reference<Task>::type&& std::move<Task>(&Task*)
[13] std::move_iterator<ClubMember>::base() const [142] std::allocator_traits<std::allocator<ClubMember*> >::allocate(std::allocator<ClubMember*>, unsigned int) [133] std::set<int>
[94] std::move_iterator<ClubMember>::operator() const [89] std::allocator_traits<std::allocator<ClubMember*> >::max_size(std::allocator<ClubMember*> const&) [15] ClubMember&& std::forward<Club
Member>::reference<ClubMember>::type()
[8] std::move_iterator<base>::base() const [119] void std::allocator_traits<std::allocator<ClubMember*> >::construct<ClubMember, ClubMember>(std::allocator<ClubMember*>, ClubMember*, ClubMember*)
std::forward<Task>(&Task)::remove_reference<Task>::type()
[43] std::move_iterator<Task*>::operator() const [120] std::allocator_traits<std::allocator<Task*> >::deallocate(std::allocator<Task*>, Task*, unsigned int) [17] int const& std::forward<int>
::reference<int> const&::type()
[25] std::move_iterator::base() const [121] std::allocator_traits<std::allocator<Task*> >::allocate(std::allocator<Task*>, unsigned int) [52] void std::destroy<ClubMember>(ClubMember*)
[7] std::Bit_reference::operator bool() const [65] std::allocator_traits<std::allocator<Task*> >::max_size(std::allocator<Task*> const&) [18] void std::destroy<Task>(&Task*)
[139] std::vector<ClubMember, std::allocator<ClubMember*> >::M_check_len(unsigned int, char const*) const [66] void std::allocator_traits<std::allocator<Task*> >::construct<Task, Task>(std::allocator<Task*>, Task&&)
[134] void std::destroy<ClubMember>(&ClubMember, ClubMember*)
[35] std::vector<ClubMember, std::allocator<ClubMember*> >::size() const [122] std::allocator_traits<std::allocator<int> >::deallocate(std::allocator<int>, int*, unsigned int) [135] void std::de
stroy<ClubMember>(&ClubMember, ClubMember*)
[88] std::vector<ClubMember, std::allocator<ClubMember*> >::max_size() const [123] std::allocator_traits<std::allocator<int> >::allocate(std::allocator<int>, unsigned int) [101] void std::de
stroy<ClubMember>(&ClubMember, ClubMember*)
[113] std::vector<Task, std::allocator<Task>> ::M_check_len(unsigned int, char const*) const [67] std::allocator_traits<std::allocator<int> >::max_size(std::allocator<int> const&) [102] void std
::destroy<Task>(&Task, Task*, std::allocator<Task*>)
[22] std::vector<Task, std::allocator<Task>> ::size() const [68] void std::allocator_traits<std::allocator<int> >::construct<int, int const&>(std::allocator<int>, int*, int const&) [103] void
std::destroy<Task>(&Task, int*, int const&)
[61] std::vector<Task, std::allocator<Task>> ::max_size() const [204] std::allocator_traits<std::allocator<unsigned long> >::deallocate(std::allocator<unsigned long>, unsigned long, unsigned
long) std::destroy<Task>(&Task, int*, int const&)
[14] std::vector<int, std::allocator<int>> ::M_check_len(unsigned int, char const*) const [205] std::allocator_traits<std::allocator<unsigned long> >::allocate(std::allocator<unsigned long>,
&ClubMember*, ClubMember*) [20] bool std::operator!=<Task*>(&Task, Task*) const& std::move_iterator<ClubMember*> const&
[26] std::vector<int, std::allocator<int>> ::size() const [206] std::Bit_iterator::base::M_incr(int) [19] bool std::operator==<Task*>(std::move_iterator<Task*> const&, std::move_iterator<Task*>, std::move
_iterator<int>, const&)
[62] std::vector<int, std::allocator<int>> ::max_size() const [2] std::Bit_iterator::base::Bit_iterator_base::Bit_iterator_base(unsigned long, unsigned int) [47] bool std::operator!=<ClubMember*>(std::move_it
erator<ClubMember*>, const&)
[171] std::allocator<ClubMember>::allocator() [90] ClubMember* std::uninitialized_copy<false>::uninit_copy<std::move_iterator<ClubMember>, ClubMember*>(&std::move_iterator<ClubMember>, Club
Member*, ClubMember*)
[172] std::allocator<ClubMember>::~allocator() [69] Task* std::uninitialized_copy<false>::uninit_copy<std::move_iterator<Task*>, Task*>(&std::move_iterator<Task*>, std::move_iterator<Task*>,
std::move_iterator<Task*>)
[173] std::allocator<Task>::~allocator() [70] int* std::uninitialized_copy<true>::uninit_copy<std::move_iterator<int>, int>(&std::move_iterator<int>, std::move_iterator<int>, int)

```

7 C++ Debugging Output:

```
PS G:\mini_Project> cd "g:\\" ; if $(?) & g++ TaskAlloter.cpp -o TaskAlloter ; ;  
TaskAlloter.cpp: In member function 'void Club::show_allotment()':  
TaskAlloter.cpp:108:36: error: expected primary-expression before '=' token  
    for (int i = 0, j = 0; i < = mem ; i++)  
                                ^  
PS G:\>
```

Figure 2: INITIAL WRONG OUTPUT ON EDITOR

```
C:\Users\hp\sj\shivansh>g++ -g TaskAlloter.cpp
C:\Users\hp\sj\shivansh>gdb a.exe
GNU gdb (GDB) 8.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-w64-mingw32".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.exe...done.
(gdb) b 108
Breakpoint 1 at 0x402cd5: file TaskAlloter.cpp, line 108.
(gdb) b main
Breakpoint 2 at 0x4015cf: file TaskAlloter.cpp, line 170.
(gdb) info breakpoints
Num      Type            Disp Enb Address      What
1        breakpoint      keep y  0x00402cd5 in Club::show_allotment() at TaskAlloter.cpp:108
2        breakpoint      keep y  0x004015cf in main() at TaskAlloter.cpp:170
(gdb) run
Starting program: C:\Users\hp\sj\shivansh\a.exe
[New Thread 34904.0x2ef0]
[New Thread 34904.0xb0ec]
[New Thread 34904.0x17bf8]
[New Thread 34904.0xef04]
[New Thread 34904.0x18688]

Thread 1 hit Breakpoint 2, main () at TaskAlloter.cpp:170
170          Club club;
(gdb) -
```

Figure 3: Program stop while running in terminal at breakpoints

```
C:\Users\hp\sj\shivansh>g++ -g TaskAlloter.cpp
C:\Users\hp\sj\shivansh>gdb a.exe
GNU gdb (GDB) 8.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY; to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-w64-mingw32".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.exe...done.
(gdb) info b
No breakpoints or watchpoints.
(gdb) b 108
Breakpoint 1 at 0x402cd5: file TaskAlloter.cpp, line 108.
(gdb) run
Starting program: C:\Users\hp\sj\shivansh\TaskAlloter.exe
[New Thread 24624.0x15820]
[New Thread 24624.0xc1f8]
[New Thread 24624.0xbbc]
[New Thread 24624.0xb418]
[New Thread 24624.0x6570]
Enter total no of members: [New Thread 24624.0x16b50]
[New Thread 24624.0xef2c]
[Thread 24624.0x6570 exited with code 0]
[Thread 24624.0x16b50 exited with code 1]
[Thread 24624.0xbbc exited with code 0]
[Thread 24624.0xb418 exited with code 0]
[Thread 24624.0xc1f8 exited with code 0]
```

Figure 4: After correcting code, program keep running without stopping at breakpoint

```
PS G:\Mini_Project> cd "g:\" ; if ($?) { g++ TaskAlloter.cpp -o TaskAlloter } ; if ($?) {  
Enter total no of members: 2  
Enter name of 1 person : abc  
Enter max tasks : 1  
Enter name of 2 person : pqr  
Enter max tasks : 2  
Enter total no of Task to be allotted : 3  
Enter task 1 name : debug  
Enter task 2 name : profile  
Enter task 3 name : latex  
    abc : debug  
    pqr : latex profile  
PS G:\> █
```

Figure 5: Correct Output on editor

8 Following are the functions used in JAVA CODE

8.1 setTaskName

This function is used to give input about Task Name .

8.2 getTaskName

This function is used to get output about Task Name.

8.3 setClubMember

This function is used to set input about Club members including task name and maximum task limit.

8.4 factorial

This function is used to find factorial.

8.5 kthPermutationUtil

This is Kth permutation utility function.

8.6 kthPermutation

This function is used to find Kth Permutation.

8.7 showallotment

This function will be called after taking all inputs to show possible allotment combination.

8.8 printClubMembers

This function is used to print club member name.

8.9 addMember

This functions will be used to add club member if required.

8.10 takeinput

This function will be used to take input about total member count, member names, task limit per person, total number of task and name of each task.

9 Code in JAVA LANGUAGE:

```

import java.util.*;

class Task
{
    String task_name;

    //default constructor
    Task(){}

    //constructor with 1 String argument
    Task(String s){                         //constructor overloading
        task_name = s;
    }

    //function to give input about Task Name
    void setTaskName(String name){
        task_name = name;
    }

    //function to get output about Task Name
    String getTaskName(){
        return task_name;
    }
}

class ClubMember
{
    String name;
    int maxTasks;

    //default constructor
    ClubMember(){}

    //constructor with 2 argument(String , integer)
    ClubMember(String n , int maxT){           //constructor overloading
        name = n;
        maxTasks = maxT;
    }

    //function to set input about Club Member
    void setClubMember(String s , int maxT){
        name = s;
        maxTasks = maxT;
    }
}

```

```

class Club
{
    private List<ClubMember> club_member;
    private List<Task> task_info;

    Club(){
        club_member = new ArrayList<ClubMember>();
        task_info = new ArrayList<Task>();
    }

    // function to find factorial
    private long factorial(int n)
    {
        int fact = 1;
        for (int i = 1; i <= n; i++){
            fact *= i;
        }
        return fact;
    }

    //Kth permutation utility function
    private void kthPermutationUtil(List<Integer> v, List<Boolean> arr, int len, int n)
    {
        if (len == 0){
            return;
        }
        long fact = factorial(len - 1);
        long x = k / fact;
        int num;
        for (num = 0; num < n; num++)
        {
            if (arr.get(num) == false && ((x--) == 0)){
                break;
            }
        }
        v.add(num);
        arr.set(num, true);
        kthPermutationUtil(v, arr, len - 1, n, k % fact);
    }

    //function to found Kth permutation
    private List<Integer> kthPermutation(int n, long k)
    {
        List<Integer> v = new ArrayList<Integer>();
        List<Boolean> arr = new ArrayList<Boolean>(n);
        for (int i=0 ; i<n ; i++){
            arr.add(i, false);
        }
    }
}

```

```

    }

    kthPermutationUtil(v, arr, n, n, k - 1);
    return v;
}

//function to show possible allotment combination
public void show_allotment()
{
    int n = task_info.size();
    int mem = club_member.size();
    Random rand = new Random();
    long k = rand.nextLong(factorial(n)) + 1;

    List<Integer> ind = kthPermutation(n, k);

    for (int i = 0, j = 0; i < mem; i++)
    {
        System.out.printf("%-10s : ", club_member.get(i).name);
        for (int l = j; l < j + club_member.get(i).maxTasks && l < n; l++){
            System.out.print(task_info.get(ind.get(l)).task_name + " ");
        }

        System.out.println("");
        j += club_member.get(i).maxTasks;
        if (j >= n){
            break;
        }
    }
}

//function to print club members name
public void printClubMembers(){
    int n = club_member.size();
    for (int i=0 ; i<n ; i++){
        System.out.print(club_member.get(i) + " ");
    }
}

//function to add club member
public void addMember(ClubMember mem){
    club_member.add(mem);
}

//function to get input about total member count, member name, task limit, total t
public void take_input()
{

```

```

Scanner sc = new Scanner(System.in);
System.out.print("Enter_total_no_of_members:");
int mem_count = sc.nextInt();
sc.nextLine();

for (int i = 0; i < mem_count; i++) {
    System.out.print("Enter_name_of_" + (i + 1) + "_person:");
    String str = sc.nextLine();
    System.out.print("Enter_max_tasks:");
    int maxTasks = sc.nextInt();
    sc.nextLine();
    club_member.add(new ClubMember(str, maxTasks));
}

System.out.print("Enter_total_no_of_Task_to_be_allotted:");
int task_count = sc.nextInt();
sc.nextLine();
for (int i = 0; i < task_count; i++) {
    System.out.print("Enter_task_" + (i + 1) + "_name:");
    String task = sc.nextLine();
    task_info.add(new Task(task));
}
sc.close();
}

public class TaskAlloter_1{
    public static void main(String[] args){
        Club club = new Club();
        club.take_input();
        club.show_allotment();
    }
}

```

10 JAVA CODE OUTPUT:

```
PS C:\Users\hp> cd "g:\JAVA\" ; if (?) { javac TaskAllotter_1.java } ; if (?) { java TaskAllotter_1
Enter total no of members : 5
Enter name of 1 person : shivansh
Enter max tasks : 2
Enter name of 2 person : tanisha
Enter max tasks : 2
Enter name of 3 person : shubhra
Enter max tasks : 2
Enter name of 4 person : saksham
Enter max tasks : 1
Enter name of 5 person : soumya
Enter max tasks : 2
Enter total no of Task to be allotted : 10
Enter task 1 name : Content_Writing
Enter task 2 name : Graphic_Designing
Enter task 3 name : Video_Editing
Enter task 4 name : Management
Enter task 5 name : Logistics
Enter task 6 name : Operations
Enter task 7 name : Social_Media_Handles
Enter task 8 name : Mentor_Permission
Enter task 9 name : Offline_Promotion
Enter task 10 name : Sponsorship_Arrangement
shivansh : Video_Editing Management
tanisha : Graphic_Designing Offline_Promotion
shubhra : Operations Social_Media_Handles
saksham : Content_Writing
soumya : Sponsorship_Arrangement Logistics
PS G:\JAVA> █
```

Figure 6: JAVA CODE OUTPUT ON VS CODE EDITOR

11 JAVA PROFILING(USING VISUALVM):

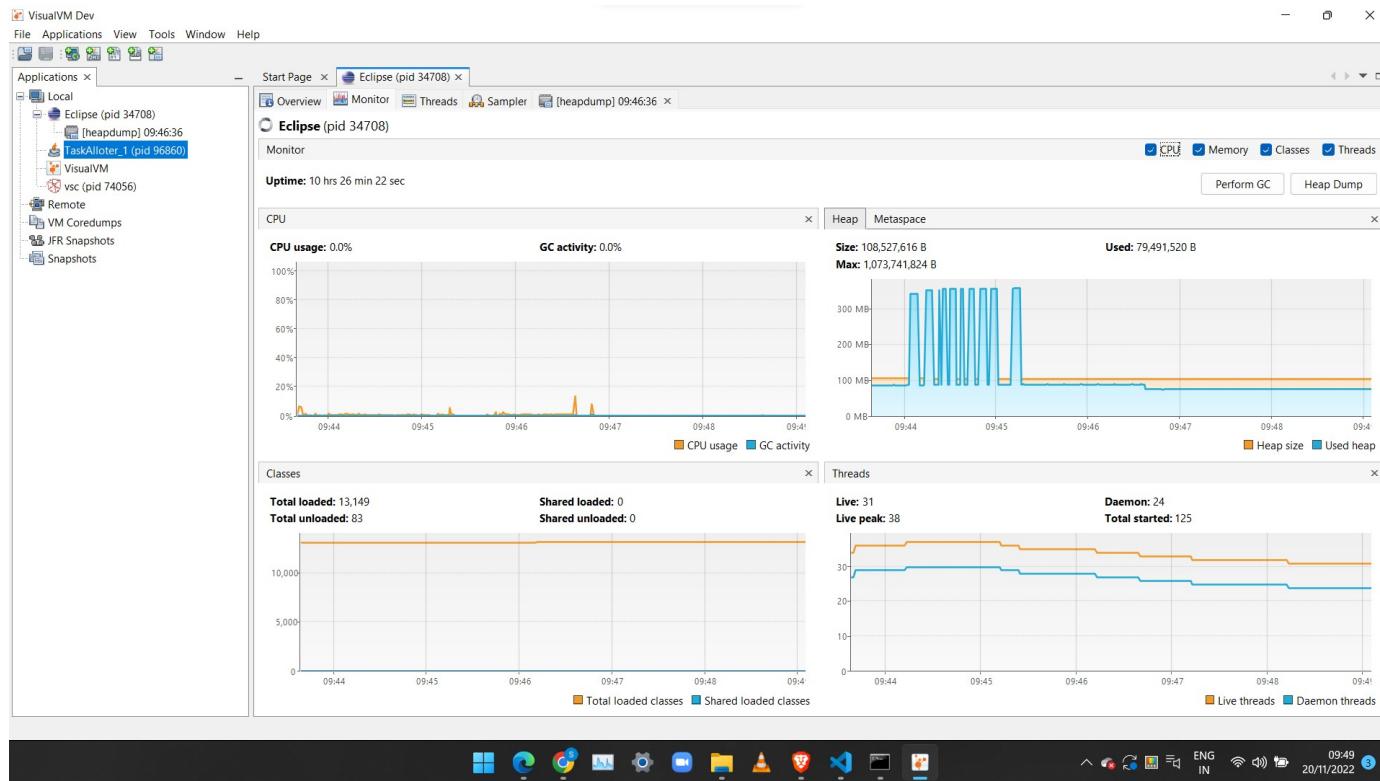


Figure 7: PROCESS EXECUTION

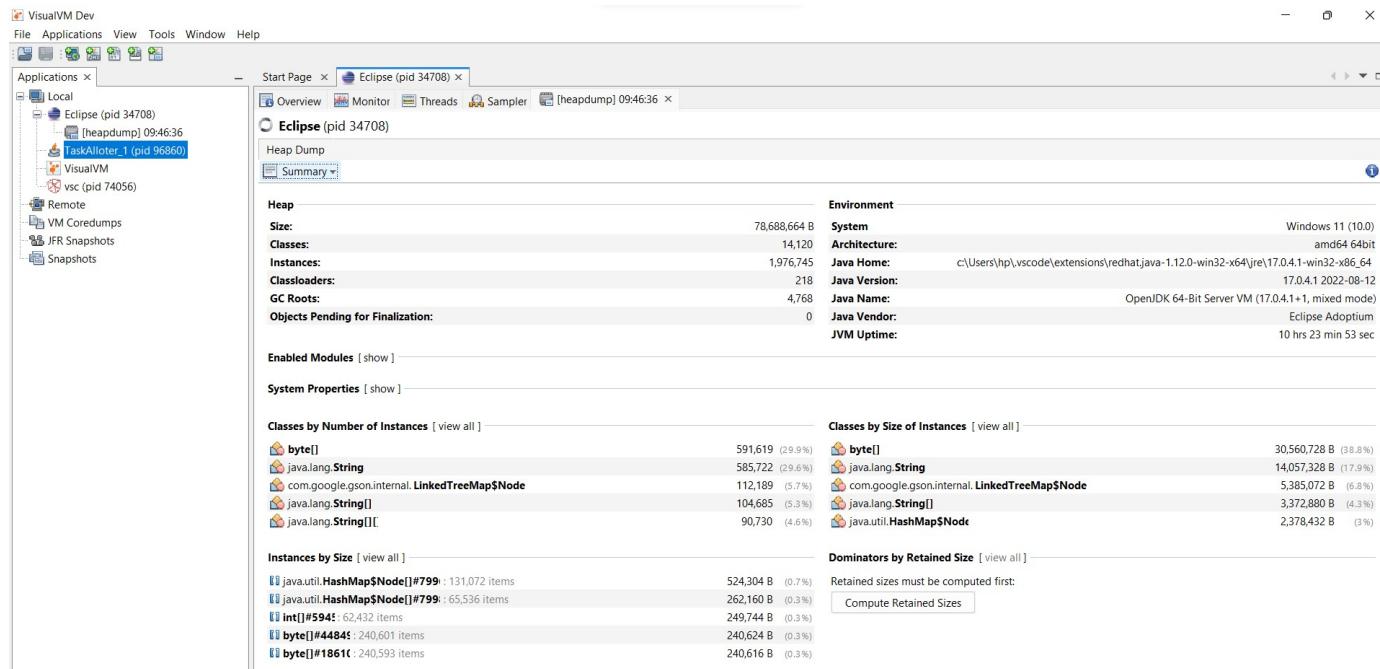


Figure 8: HEAPDUMP

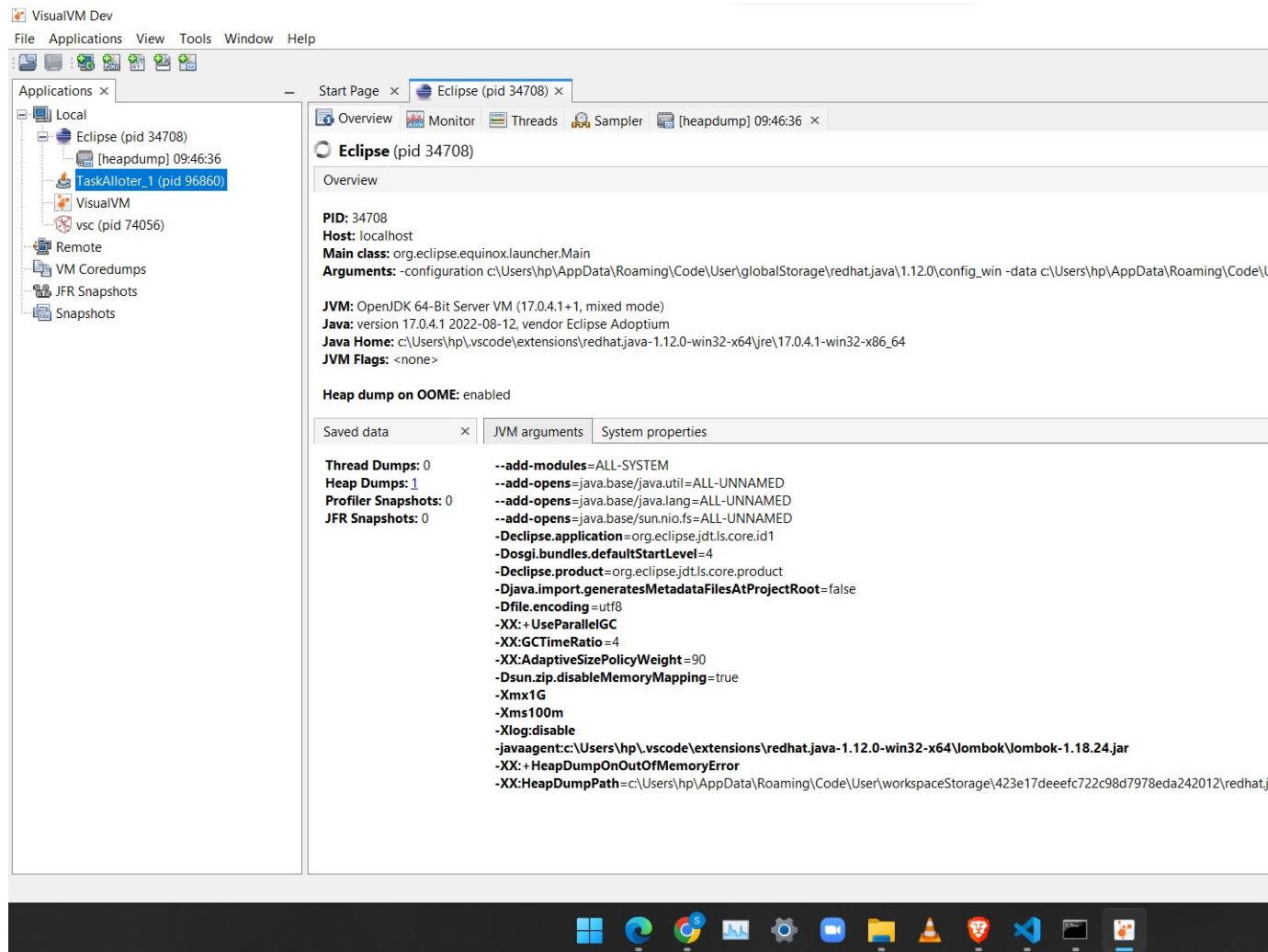


Figure 9: PROGRAM OVERVIEW

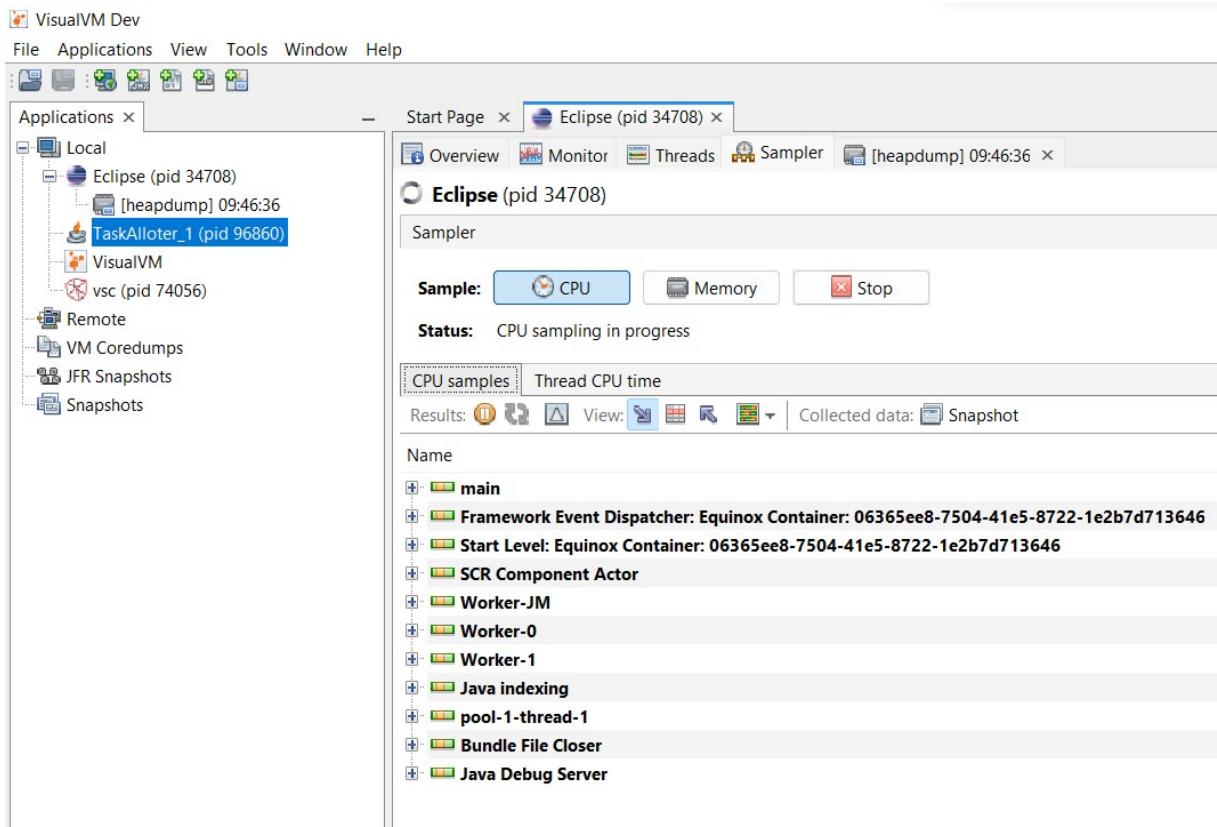


Figure 10: CPU STATUS

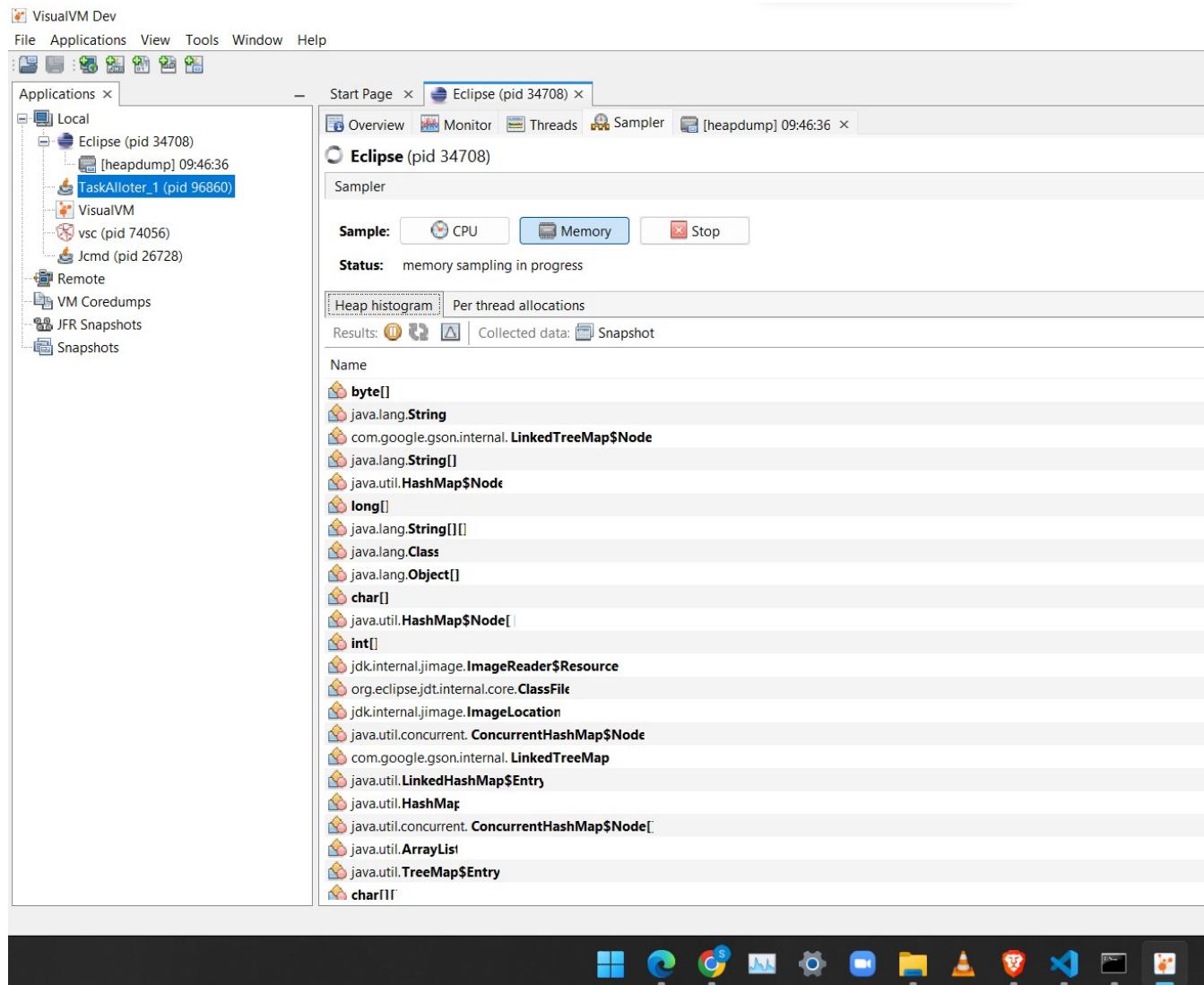


Figure 11: MEMORY STATUS

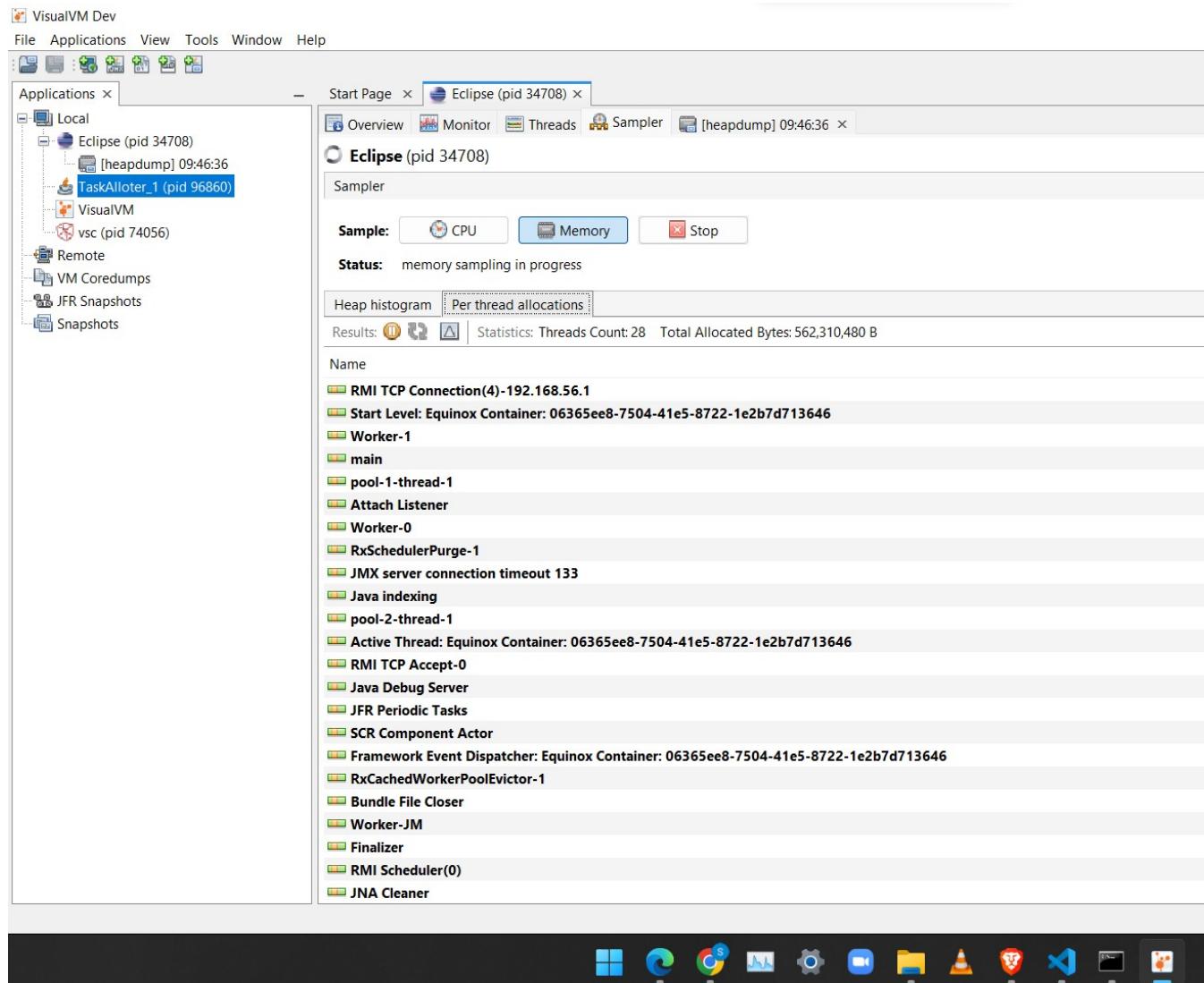


Figure 12: Thread Allocation

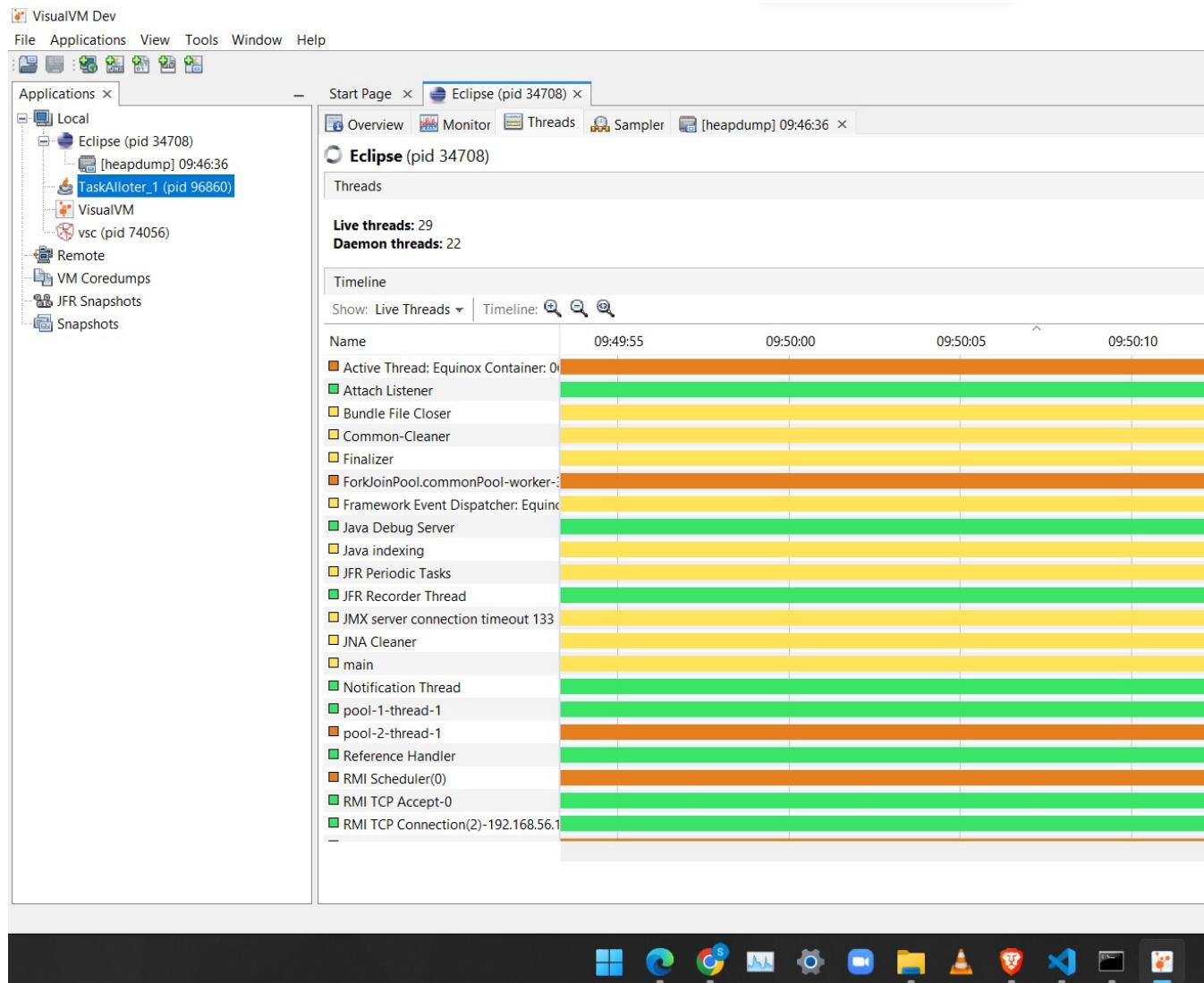


Figure 13: Thread Allocation 1

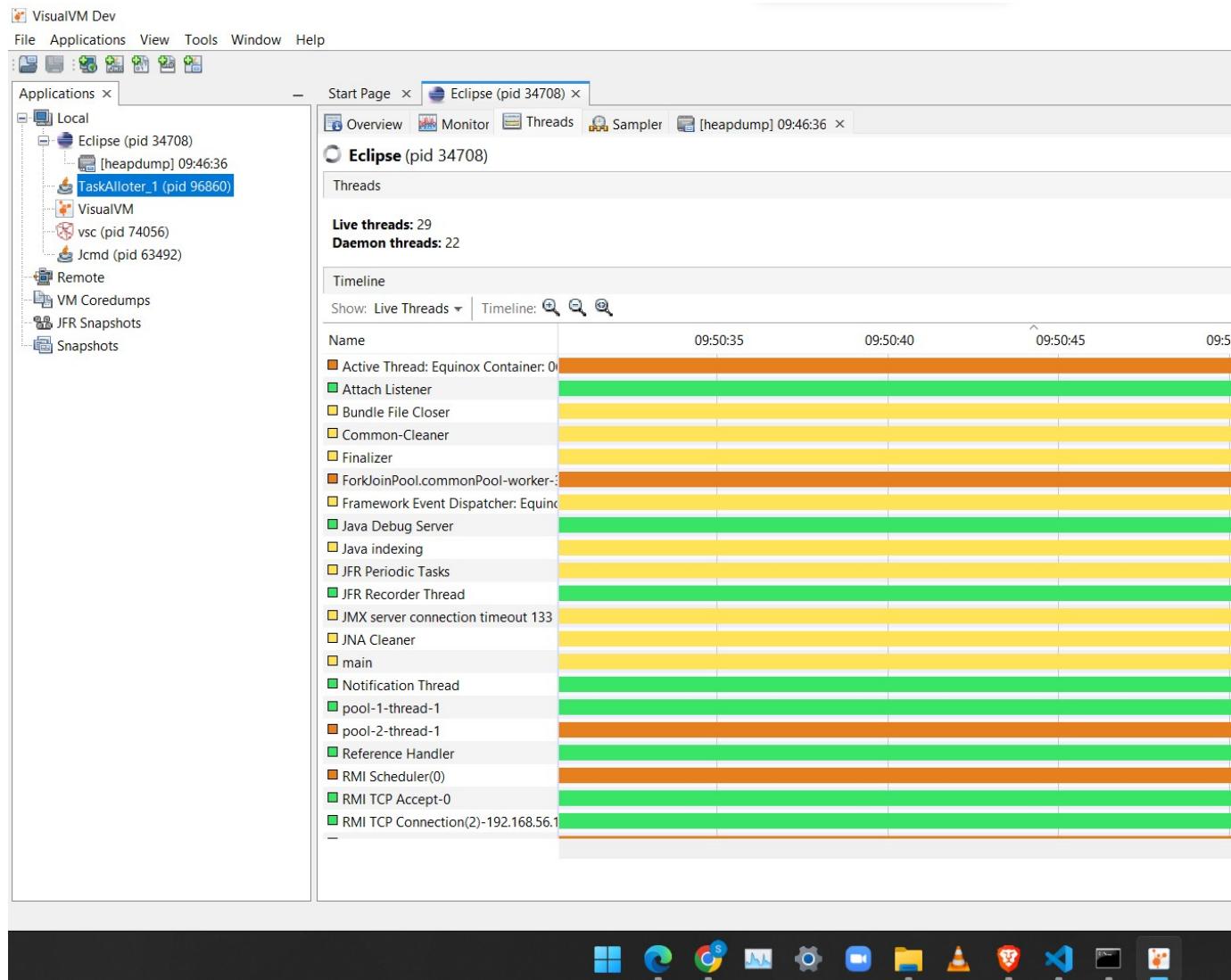


Figure 14: Thread Allocation 2

12 JAVA DEBUGGING:



Figure 15: INCORRECT CODE

```

PS G:\Mini_Project> & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe' '-agentlib:jdwp=transport=dt_tty,server=y,address=localhost:50511' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\hp\AppData\Roaming\423e17deeffc722c98d7978eda242012\redhat.java\jdt_ws\jdt_ls-java-project\bin' 'TaskAllotter_1'
Enter total no of members : 3
Enter name of 1 person : shivansh
Enter max tasks : 2
Enter name of 2 person : saksham
Enter max tasks : 1
Enter name of 3 person : nihal
Enter max tasks : 1
Enter total no of Task to be allotted : 5
Enter task 1 name : content
Enter task 2 name : media
Enter task 3 name : graphic
Enter task 4 name : management
Enter task 5 name : promotion
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    This method must return a result of type List<Integer>
        at Club.kthPermutation(TaskAllotter_1.java:89)
        at Club.show_allotment(TaskAllotter_1.java:109)
        at TaskAllotter_1.main(TaskAllotter_1.java:172)
PS G:\Mini_Project>

```

Figure 16: WRONG CODE OUTPUT: SHOWING ERROR ON TERMINAL

```
PS G:\Mini_Project> cd "g:\JAVA\" ; if (?) { javac TaskAllotter_1.java } ; if (?) { java Task
Enter total no of members : 3
Enter name of 1 person : shivansh
Enter max tasks : 2
Enter name of 2 person : saksham
Enter max tasks : 2
Enter name of 3 person : nihal
Enter max tasks : 1
Enter total no of Task to be allotted : 5
Enter task 1 name : media
Enter task 2 name : content
Enter task 3 name : graphic
Enter task 4 name : management
Enter task 5 name : sponsor_arrangement
shivansh    : content media
saksham    : graphic management
nihal      : sponsor_arrangement
PS G:\JAVA> █
```

Figure 17: CORRECT CODE OUTPUT AFTER DEBUGGING AND CORRECTING CODE