2. PROFILING OUTPUT

NAME – ABHISHEK VITHAL KARATAGI
PRN – 22610034

Flat profile:

```
Each sample counts as 0.01 seconds.
 no time accumulated

  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00      0.00     0.00     1010     0.00     0.00  __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::base() const
  0.00      0.00     0.00      505     0.00     0.00  bool __gnu_cxx::operator!=<int*, std::vector<int, std::allocator<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&)
  0.00      0.00     0.00      500     0.00     0.00  __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::operator++()
  0.00      0.00     0.00      500     0.00     0.00  __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::operator*() const
  0.00      0.00     0.00      219     0.00     0.00  std::vector<int, std::allocator<int> >::operator[](unsigned long)
  0.00      0.00     0.00      202     0.00     0.00  __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::base() const
  0.00      0.00     0.00      199     0.00     0.00  __gnu_cxx::__enable_if<std::__is_integer<int>::__value, double>::__type std::sqrt<int>(int)
  0.00      0.00     0.00      102     0.00     0.00  std::vector<int, std::allocator<int> >::size() const
  0.00      0.00     0.00      101     0.00     0.00  bool __gnu_cxx::operator!=<int const*, std::vector<int, std::allocator<int> >
>(__gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > > const&, __gnu_cxx::__normal_iterator<int const*,
std::vector<int, std::allocator<int> > > const&)
  0.00      0.00     0.00      100     0.00     0.00  isEven(int)
  0.00      0.00     0.00      100     0.00     0.00  isPrime(int)
  0.00      0.00     0.00      100     0.00     0.00  __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::operator++()
  0.00      0.00     0.00      100     0.00     0.00  __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::operator*() const
  0.00      0.00     0.00      100     0.00     0.00  std::vector<int, std::allocator<int> >::operator[](unsigned long) const
  0.00      0.00     0.00       10     0.00     0.00  factorial(int)
  0.00      0.00     0.00       10     0.00     0.00  __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::__normal_iterator(int*
const&)
  0.00      0.00     0.00        6     0.00     0.00  std::__new_allocator<int>::~__new_allocator()
  0.00      0.00     0.00        5     0.00     0.00  std::vector<int, std::allocator<int> >::end()
  0.00      0.00     0.00        5     0.00     0.00  std::vector<int, std::allocator<int> >::begin()
  0.00      0.00     0.00        4     0.00     0.00  std::_Vector_base<int, std::allocator<int> >::_M_get_Tp_allocator()
  0.00      0.00     0.00        2     0.00     0.00  __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::__normal_iterator(int
const* const&)
  0.00      0.00     0.00        2     0.00     0.00  void std::_Destroy_aux<true>::__destroy<int*>(int*, int*)
  0.00      0.00     0.00        2     0.00     0.00  std::_Vector_base<int, std::allocator<int> >::_M_allocate(unsigned long)
  0.00      0.00     0.00        2     0.00     0.00  std::_Vector_base<int, std::allocator<int> >::_Vector_impl::_Vector_impl(std::allocator<int> const&)
  0.00      0.00     0.00        2     0.00     0.00  std::_Vector_base<int, std::allocator<int> >::_Vector_impl::~_Vector_impl()
  0.00      0.00     0.00        2     0.00     0.00  std::_Vector_base<int, std::allocator<int> >::_M_deallocate(int*, unsigned long)
  0.00      0.00     0.00        2     0.00     0.00  std::_Vector_base<int, std::allocator<int> >::_M_create_storage(unsigned long)
  0.00      0.00     0.00        2     0.00     0.00  std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_Vector_impl_data()
  0.00      0.00     0.00        2     0.00     0.00  std::_Vector_base<int, std::allocator<int> >::_Vector_base(unsigned long, std::allocator<int> const&)
  0.00      0.00     0.00        2     0.00     0.00  std::_Vector_base<int, std::allocator<int> >::~_Vector_base()
  0.00      0.00     0.00        2     0.00     0.00  std::__new_allocator<int>::deallocate(int*, unsigned long)
  0.00      0.00     0.00        2     0.00     0.00  std::__new_allocator<int>::allocate(unsigned long, void const*)
  0.00      0.00     0.00        2     0.00     0.00  int* std::__uninitialized_default_n_1<true>::__uninit_default_n<int*, unsigned long>(int*, unsigned
long)
  0.00      0.00     0.00        2     0.00     0.00  std::vector<int, std::allocator<int> >::_S_max_size(std::allocator<int> const&)
  0.00      0.00     0.00        2     0.00     0.00  std::vector<int, std::allocator<int> >::_S_check_init_len(unsigned long, std::allocator<int> const&)
  0.00      0.00     0.00        2     0.00     0.00  std::vector<int, std::allocator<int> >::_M_default_initialize(unsigned long)
  0.00      0.00     0.00        2     0.00     0.00  std::vector<int, std::allocator<int> >::vector(unsigned long, std::allocator<int> const&)
  0.00      0.00     0.00        2     0.00     0.00  std::vector<int, std::allocator<int> >::~vector()
  0.00      0.00     0.00        2     0.00     0.00  void std::_Construct<int>(int*)
  0.00      0.00     0.00        2     0.00     0.00  int* std::__fill_n_a<int*, unsigned long, int>(int*, unsigned long, int const&,
std::random_access_iterator_tag)
  0.00      0.00     0.00        2     0.00     0.00  int* std::__addressof<int>(int&)
  0.00      0.00     0.00        2     0.00     0.00  std::__size_to_integer(unsigned long)
  0.00      0.00     0.00        2     0.00     0.00  int* std::__uninitialized_default_n<int*, unsigned long>(int*, unsigned long)
  0.00      0.00     0.00        2     0.00     0.00  int* std::__uninitialized_default_n_a<int*, unsigned long, int>(int*, unsigned long, std::allocator<int>&)
  0.00      0.00     0.00        2     0.00     0.00  unsigned long const& std::min<unsigned long>(unsigned long const&, unsigned long const&)
  0.00      0.00     0.00        2     0.00     0.00  int* std::fill_n<int*, unsigned long, int>(int*, unsigned long, int const&)
  0.00      0.00     0.00        2     0.00     0.00  void std::_Destroy<int*>(int*, int*)
  0.00      0.00     0.00        2     0.00     0.00  void std::__fill_a<int*, int>(int*, int*, int const&)
  0.00      0.00     0.00        2     0.00     0.00  __gnu_cxx::__enable_if<std::__is_scalar<int>::__value, void>::__type std::__fill_a1<int*, int>(int*, int*,
int const&)
  0.00      0.00     0.00        2     0.00     0.00  operator new(unsigned long, void*)
  0.00      0.00     0.00        1     0.00     0.00  computeAverage(std::vector<int, std::allocator<int> > const&)
  0.00      0.00     0.00        1     0.00     0.00  generateRandomNumbers(std::vector<int, std::allocator<int> >&)
  0.00      0.00     0.00        1     0.00     0.00  prefixSum(std::vector<int, std::allocator<int> > const&)
  0.00      0.00     0.00        1     0.00     0.00  std::vector<int, std::allocator<int> >::end() const
  0.00      0.00     0.00        1     0.00     0.00  std::vector<int, std::allocator<int> >::begin() const

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
                   else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
                   function is profiled, else blank.

name       the name of the function.  This is the minor sort
           for this listing. The index shows the location of
                   the function in the gprof listing. If the index is
                   in parenthesis it shows where it would appear in
                   the gprof listing if it were to be printed.Copyright (C) 2012-2024 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.

                                     Call graph (explanation follows)


granularity: each sample hit covers 2 byte(s) no time propagated

index % time    self  children    called     name
                0.00    0.00    1010/1010        bool __gnu_cxx::operator!=<int*, std::vector<int, std::allocator<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&) [9]
[8]      0.0    0.00    0.00    1010         __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::base() const [8]
-----------------------------------------------
                0.00    0.00     101/505         generateRandomNumbers(std::vector<int, std::allocator<int> >&) [59]
                0.00    0.00     404/505         main [6]
[9]      0.0    0.00    0.00     505         bool __gnu_cxx::operator!=<int*, std::vector<int, std::allocator<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&) [9]
                0.00    0.00    1010/1010        __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::base() const [8]
-----------------------------------------------
                0.00    0.00     100/500         generateRandomNumbers(std::vector<int, std::allocator<int> >&) [59]
                0.00    0.00     400/500         main [6]
[10]     0.0    0.00    0.00     500         __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::operator++() [10]
-----------------------------------------------
                0.00    0.00     100/500         generateRandomNumbers(std::vector<int, std::allocator<int> >&) [59]
                0.00    0.00     400/500         main [6]
[11]     0.0    0.00    0.00     500         __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::operator*() const [11]
-----------------------------------------------
                0.00    0.00      20/219         main [6]
                0.00    0.00     199/219         prefixSum(std::vector<int, std::allocator<int> > const&) [60]
[12]     0.0    0.00    0.00     219         std::vector<int, std::allocator<int> >::operator[](unsigned long) [12]
-----------------------------------------------
                0.00    0.00     202/202         bool __gnu_cxx::operator!=<int const*, std::vector<int, std::allocator<int> >
>(__gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > > const&, __gnu_cxx::__normal_iterator<int const*,
std::vector<int, std::allocator<int> > > const&) [16]
[13]     0.0    0.00    0.00     202         __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::base() const [13]
-----------------------------------------------
                0.00    0.00     199/199         isPrime(int) [18]
[14]     0.0    0.00    0.00     199         __gnu_cxx::__enable_if<std::__is_integer<int>::__value, double>::__type std::sqrt<int>(int) [14]
-----------------------------------------------
                0.00    0.00       1/102         computeAverage(std::vector<int, std::allocator<int> > const&) [58]
                0.00    0.00     101/102         prefixSum(std::vector<int, std::allocator<int> > const&) [60]
[15]     0.0    0.00    0.00     102         std::vector<int, std::allocator<int> >::size() const [15]
-----------------------------------------------
                0.00    0.00     101/101         computeAverage(std::vector<int, std::allocator<int> > const&) [58]
[16]     0.0    0.00    0.00     101         bool __gnu_cxx::operator!=<int const*, std::vector<int, std::allocator<int> >
>(__gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > > const&, __gnu_cxx::__normal_iterator<int const*,
std::vector<int, std::allocator<int> > > const&) [16]
                0.00    0.00     202/202         __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::base() const [13]
```

```
-----------------------------------------------
                0.00    0.00     100/100         main [6]
[17]     0.0    0.00    0.00     100         isEven(int) [17]
-----------------------------------------------
                0.00    0.00     100/100         main [6]
[18]     0.0    0.00    0.00     100         isPrime(int) [18]
                0.00    0.00     199/199         __gnu_cxx::__enable_if<std::__is_integer<int>::__value, double>::__type std::sqrt<int>(int) [14]
-----------------------------------------------
                0.00    0.00     100/100         computeAverage(std::vector<int, std::allocator<int> > const&) [58]
[19]     0.0    0.00    0.00     100         __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::operator++() [19]
-----------------------------------------------
                0.00    0.00     100/100         computeAverage(std::vector<int, std::allocator<int> > const&) [58]
[20]     0.0    0.00    0.00     100         __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::operator*() const [20]
-----------------------------------------------
                0.00    0.00     100/100         prefixSum(std::vector<int, std::allocator<int> > const&) [60]
[21]     0.0    0.00    0.00     100         std::vector<int, std::allocator<int> >::operator[](unsigned long) const [21]
-----------------------------------------------
                             56622             factorial(int) [22]
                0.00    0.00      10/10          main [6]
[22]     0.0    0.00    0.00      10+56622   factorial(int) [22]
                             56622             factorial(int) [22]
-----------------------------------------------
                0.00    0.00       5/10          std::vector<int, std::allocator<int> >::begin() [26]
                0.00    0.00       5/10          std::vector<int, std::allocator<int> >::end() [25]
[23]     0.0    0.00    0.00      10         __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::__normal_iterator(int* const&) [23]
-----------------------------------------------
                0.00    0.00       1/6           main [6]
                0.00    0.00       1/6           prefixSum(std::vector<int, std::allocator<int> > const&) [60]
                0.00    0.00       2/6           std::_Vector_base<int, std::allocator<int> >::_Vector_impl::~_Vector_impl() [32]
                0.00    0.00       2/6           std::vector<int, std::allocator<int> >::_S_check_init_len(unsigned long, std::allocator<int> const&) [42]
[24]     0.0    0.00    0.00       6         std::__new_allocator<int>::~__new_allocator() [24]
-----------------------------------------------
                0.00    0.00       1/5           generateRandomNumbers(std::vector<int, std::allocator<int> >&) [59]
                0.00    0.00       4/5           main [6]
[25]     0.0    0.00    0.00       5         std::vector<int, std::allocator<int> >::end() [25]
                0.00    0.00       5/10          __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::__normal_iterator(int* const&) [23]
-----------------------------------------------
                0.00    0.00       1/5           generateRandomNumbers(std::vector<int, std::allocator<int> >&) [59]
                0.00    0.00       4/5           main [6]
[26]     0.0    0.00    0.00       5         std::vector<int, std::allocator<int> >::begin() [26]
                0.00    0.00       5/10          __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::__normal_iterator(int* const&) [23]
-----------------------------------------------
                0.00    0.00       2/4           std::vector<int, std::allocator<int> >::~vector() [45]
                0.00    0.00       2/4           std::vector<int, std::allocator<int> >::_M_default_initialize(unsigned long) [43]
[27]     0.0    0.00    0.00       4         std::_Vector_base<int, std::allocator<int> >::_M_get_Tp_allocator() [27]
-----------------------------------------------
                0.00    0.00       1/2           std::vector<int, std::allocator<int> >::begin() const [62]
                0.00    0.00       1/2           std::vector<int, std::allocator<int> >::end() const [61]
[28]     0.0    0.00    0.00       2         __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::__normal_iterator(int const*
const&) [28]
-----------------------------------------------
                0.00    0.00       2/2           void std::_Destroy<int*>(int*, int*) [54]
[29]     0.0    0.00    0.00       2         void std::_Destroy_aux<true>::__destroy<int*>(int*, int*) [29]
-----------------------------------------------
                0.00    0.00       2/2           std::_Vector_base<int, std::allocator<int> >::_M_create_storage(unsigned long) [34]
[30]     0.0    0.00    0.00       2         std::_Vector_base<int, std::allocator<int> >::_M_allocate(unsigned long) [30]
                0.00    0.00       2/2           std::__new_allocator<int>::allocate(unsigned long, void const*) [39]
-----------------------------------------------
                0.00    0.00       2/2           std::_Vector_base<int, std::allocator<int> >::_Vector_base(unsigned long, std::allocator<int> const&) [36]
[31]     0.0    0.00    0.00       2         std::_Vector_base<int, std::allocator<int> >::_Vector_impl::_Vector_impl(std::allocator<int> const&) [31]
                0.00    0.00       2/2           std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_Vector_impl_data() [35]
-----------------------------------------------
                0.00    0.00       2/2           std::_Vector_base<int, std::allocator<int> >::~_Vector_base() [37]
[32]     0.0    0.00    0.00       2         std::_Vector_base<int, std::allocator<int> >::_Vector_impl::~_Vector_impl() [32]
                0.00    0.00       2/6           std::__new_allocator<int>::~__new_allocator() [24]
-----------------------------------------------
                0.00    0.00       2/2           std::_Vector_base<int, std::allocator<int> >::~_Vector_base() [37]
[33]     0.0    0.00    0.00       2         std::_Vector_base<int, std::allocator<int> >::_M_deallocate(int*, unsigned long) [33]
                0.00    0.00       2/2           std::__new_allocator<int>::deallocate(int*, unsigned long) [38]
-----------------------------------------------
                0.00    0.00       2/2           std::_Vector_base<int, std::allocator<int> >::_Vector_base(unsigned long, std::allocator<int> const&) [36]
[34]     0.0    0.00    0.00       2         std::_Vector_base<int, std::allocator<int> >::_M_create_storage(unsigned long) [34]
                0.00    0.00       2/2           std::_Vector_base<int, std::allocator<int> >::_M_allocate(unsigned long) [30]
-----------------------------------------------
                0.00    0.00       2/2           std::_Vector_base<int, std::allocator<int> >::_Vector_impl::_Vector_impl(std::allocator<int> const&) [31]
[35]     0.0    0.00    0.00       2         std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_Vector_impl_data() [35]
-----------------------------------------------
                0.00    0.00       2/2           std::vector<int, std::allocator<int> >::vector(unsigned long, std::allocator<int> const&) [44]
[36]     0.0    0.00    0.00       2         std::_Vector_base<int, std::allocator<int> >::_Vector_base(unsigned long, std::allocator<int> const&) [36]
```

```
                0.00    0.00       2/2           std::_Vector_base<int, std::allocator<int> >::_Vector_impl::_Vector_impl(std::allocator<int> const&) [31]
                0.00    0.00       2/2           std::_Vector_base<int, std::allocator<int> >::_M_create_storage(unsigned long) [34]
-----------------------------------------------
                0.00    0.00       2/2           std::vector<int, std::allocator<int> >::~vector() [45]
[37]     0.0    0.00    0.00       2         std::_Vector_base<int, std::allocator<int> >::~_Vector_base() [37]
                0.00    0.00       2/2           std::_Vector_base<int, std::allocator<int> >::_Vector_impl::~_Vector_impl() [32]
                0.00    0.00       2/2           std::_Vector_base<int, std::allocator<int> >::_M_deallocate(int*, unsigned long) [33]
-----------------------------------------------
                0.00    0.00       2/2           std::_Vector_base<int, std::allocator<int> >::_M_deallocate(int*, unsigned long) [33]
[38]     0.0    0.00    0.00       2         std::__new_allocator<int>::deallocate(int*, unsigned long) [38]
-----------------------------------------------
                0.00    0.00       2/2           std::_Vector_base<int, std::allocator<int> >::_M_allocate(unsigned long) [30]
[39]     0.0    0.00    0.00       2         std::__new_allocator<int>::allocate(unsigned long, void const*) [39]
-----------------------------------------------
                0.00    0.00       2/2           int* std::__uninitialized_default_n<int*, unsigned long>(int*, unsigned long) [50]
[40]     0.0    0.00    0.00       2         int* std::__uninitialized_default_n_1<true>::__uninit_default_n<int*, unsigned long>(int*, unsigned long) [40]
                0.00    0.00       2/2           int* std::__addressof<int>(int&) [48]
                0.00    0.00       2/2           void std::_Construct<int>(int*) [46]
                0.00    0.00       2/2           int* std::fill_n<int*, unsigned long, int>(int*, unsigned long, int const&) [53]
-----------------------------------------------
                0.00    0.00       2/2           std::vector<int, std::allocator<int> >::_S_check_init_len(unsigned long, std::allocator<int> const&) [42]
[41]     0.0    0.00    0.00       2         std::vector<int, std::allocator<int> >::_S_max_size(std::allocator<int> const&) [41]
                0.00    0.00       2/2           unsigned long const& std::min<unsigned long>(unsigned long const&, unsigned long const&) [52]
-----------------------------------------------
                0.00    0.00       2/2           std::vector<int, std::allocator<int> >::vector(unsigned long, std::allocator<int> const&) [44]
[42]     0.0    0.00    0.00       2         std::vector<int, std::allocator<int> >::_S_check_init_len(unsigned long, std::allocator<int> const&) [42]
                0.00    0.00       2/2           std::vector<int, std::allocator<int> >::_S_max_size(std::allocator<int> const&) [41]
                0.00    0.00       2/6           std::__new_allocator<int>::~__new_allocator() [24]
-----------------------------------------------
                0.00    0.00       2/2           std::vector<int, std::allocator<int> >::vector(unsigned long, std::allocator<int> const&) [44]
[43]     0.0    0.00    0.00       2         std::vector<int, std::allocator<int> >::_M_default_initialize(unsigned long) [43]
                0.00    0.00       2/4           std::_Vector_base<int, std::allocator<int> >::_M_get_Tp_allocator() [27]
                0.00    0.00       2/2           int* std::__uninitialized_default_n_a<int*, unsigned long, int>(int*, unsigned long, std::allocator<int>&) [51]
-----------------------------------------------
                0.00    0.00       1/2           main [6]
                0.00    0.00       1/2           prefixSum(std::vector<int, std::allocator<int> > const&) [60]
[44]     0.0    0.00    0.00       2         std::vector<int, std::allocator<int> >::vector(unsigned long, std::allocator<int> const&) [44]
                0.00    0.00       2/2           std::vector<int, std::allocator<int> >::_S_check_init_len(unsigned long, std::allocator<int> const&) [42]
                0.00    0.00       2/2           std::_Vector_base<int, std::allocator<int> >::_Vector_base(unsigned long, std::allocator<int> const&) [36]
                0.00    0.00       2/2           std::vector<int, std::allocator<int> >::_M_default_initialize(unsigned long) [43]
-----------------------------------------------
                0.00    0.00       2/2           main [6]
[45]     0.0    0.00    0.00       2         std::vector<int, std::allocator<int> >::~vector() [45]
                0.00    0.00       2/4           std::_Vector_base<int, std::allocator<int> >::_M_get_Tp_allocator() [27]
                0.00    0.00       2/2           void std::_Destroy<int*>(int*, int*) [54]
                0.00    0.00       2/2           std::_Vector_base<int, std::allocator<int> >::~_Vector_base() [37]
-----------------------------------------------
                0.00    0.00       2/2           int* std::__uninitialized_default_n_1<true>::__uninit_default_n<int*, unsigned long>(int*, unsigned long) [40]
[46]     0.0    0.00    0.00       2         void std::_Construct<int>(int*) [46]
                0.00    0.00       2/2           operator new(unsigned long, void*) [57]
-----------------------------------------------
                0.00    0.00       2/2           int* std::fill_n<int*, unsigned long, int>(int*, unsigned long, int const&) [53]
[47]     0.0    0.00    0.00       2         int* std::__fill_n_a<int*, unsigned long, int>(int*, unsigned long, int const&, std::random_access_iterator_tag)
[47]
                0.00    0.00       2/2           void std::__fill_a<int*, int>(int*, int*, int const&) [55]
-----------------------------------------------
                0.00    0.00       2/2           int* std::__uninitialized_default_n_1<true>::__uninit_default_n<int*, unsigned long>(int*, unsigned long) [40]
[48]     0.0    0.00    0.00       2         int* std::__addressof<int>(int&) [48]
-----------------------------------------------
                0.00    0.00       2/2           int* std::fill_n<int*, unsigned long, int>(int*, unsigned long, int const&) [53]
[49]     0.0    0.00    0.00       2         std::__size_to_integer(unsigned long) [49]
-----------------------------------------------
                0.00    0.00       2/2           int* std::__uninitialized_default_n_a<int*, unsigned long, int>(int*, unsigned long, std::allocator<int>&) [51]
[50]     0.0    0.00    0.00       2         int* std::__uninitialized_default_n<int*, unsigned long>(int*, unsigned long) [50]
                0.00    0.00       2/2           int* std::__uninitialized_default_n_1<true>::__uninit_default_n<int*, unsigned long>(int*, unsigned long) [40]
-----------------------------------------------
                0.00    0.00       2/2           std::vector<int, std::allocator<int> >::_M_default_initialize(unsigned long) [43]
[51]     0.0    0.00    0.00       2         int* std::__uninitialized_default_n_a<int*, unsigned long, int>(int*, unsigned long, std::allocator<int>&) [51]
                0.00    0.00       2/2           int* std::__uninitialized_default_n<int*, unsigned long>(int*, unsigned long) [50]
-----------------------------------------------
                0.00    0.00       2/2           std::vector<int, std::allocator<int> >::_S_max_size(std::allocator<int> const&) [41]
[52]     0.0    0.00    0.00       2         unsigned long const& std::min<unsigned long>(unsigned long const&, unsigned long const&) [52]
-----------------------------------------------
                0.00    0.00       2/2           int* std::__uninitialized_default_n_1<true>::__uninit_default_n<int*, unsigned long>(int*, unsigned long) [40]
[53]     0.0    0.00    0.00       2         int* std::fill_n<int*, unsigned long, int>(int*, unsigned long, int const&) [53]
                0.00    0.00       2/2           std::__size_to_integer(unsigned long) [49]
                0.00    0.00       2/2           int* std::__fill_n_a<int*, unsigned long, int>(int*, unsigned long, int const&, std::random_access_iterator_tag)
[47]
-----------------------------------------------
```

```
                0.00    0.00       2/2          std::vector<int, std::allocator<int> >::~vector() [45]
[54]     0.0    0.00    0.00       2        void std::_Destroy<int*>(int*, int*) [54]
                0.00    0.00       2/2          void std::_Destroy_aux<true>::__destroy<int*>(int*, int*) [29]
-----------------------------------------------
                0.00    0.00       2/2          int* std::__fill_n_a<int*, unsigned long, int>(int*, unsigned long, int const&, std::random_access_iterator_tag)
[47]
[55]     0.0    0.00    0.00       2        void std::__fill_a<int*, int>(int*, int*, int const&) [55]
                0.00    0.00       2/2          __gnu_cxx::__enable_if<std::__is_scalar<int>::__value, void>::__type std::__fill_a1<int*, int>(int*, int*, int
const&) [56]
-----------------------------------------------
                0.00    0.00       2/2          void std::__fill_a<int*, int>(int*, int*, int const&) [55]
[56]     0.0    0.00    0.00       2        __gnu_cxx::__enable_if<std::__is_scalar<int>::__value, void>::__type std::__fill_a1<int*, int>(int*, int*, int
const&) [56]
-----------------------------------------------
                0.00    0.00       2/2          void std::_Construct<int>(int*) [46]
[57]     0.0    0.00    0.00       2        operator new(unsigned long, void*) [57]
-----------------------------------------------
                0.00    0.00       1/1          main [6]
[58]     0.0    0.00    0.00       1        computeAverage(std::vector<int, std::allocator<int> > const&) [58]
                0.00    0.00     101/101         bool __gnu_cxx::operator!=<int const*, std::vector<int, std::allocator<int> >
>(__gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > > const&, __gnu_cxx::__normal_iterator<int const*,
std::vector<int, std::allocator<int> > > const&) [16]
                0.00    0.00     100/100         __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::operator*() const [20]
                0.00    0.00     100/100         __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::operator++() [19]
                0.00    0.00       1/1          std::vector<int, std::allocator<int> >::begin() const [62]
                0.00    0.00       1/1          std::vector<int, std::allocator<int> >::end() const [61]
                0.00    0.00       1/102         std::vector<int, std::allocator<int> >::size() const [15]
-----------------------------------------------
                0.00    0.00       1/1          main [6]
[59]     0.0    0.00    0.00       1        generateRandomNumbers(std::vector<int, std::allocator<int> >&) [59]
                0.00    0.00     101/505         bool __gnu_cxx::operator!=<int*, std::vector<int, std::allocator<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&) [9]
                0.00    0.00     100/500         __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::operator*() const [11]
                0.00    0.00     100/500         __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::operator++() [10]
                0.00    0.00       1/5          std::vector<int, std::allocator<int> >::begin() [26]
                0.00    0.00       1/5          std::vector<int, std::allocator<int> >::end() [25]
-----------------------------------------------
                0.00    0.00       1/1          main [6]
[60]     0.0    0.00    0.00       1        prefixSum(std::vector<int, std::allocator<int> > const&) [60]
                0.00    0.00     199/219         std::vector<int, std::allocator<int> >::operator[](unsigned long) [12]
                0.00    0.00     101/102         std::vector<int, std::allocator<int> >::size() const [15]
                0.00    0.00     100/100         std::vector<int, std::allocator<int> >::operator[](unsigned long) const [21]
                0.00    0.00       1/6          std::__new_allocator<int>::~__new_allocator() [24]
                0.00    0.00       1/2          std::vector<int, std::allocator<int> >::vector(unsigned long, std::allocator<int> const&) [44]
-----------------------------------------------
                0.00    0.00       1/1          computeAverage(std::vector<int, std::allocator<int> > const&) [58]
[61]     0.0    0.00    0.00       1        std::vector<int, std::allocator<int> >::end() const [61]
                0.00    0.00       1/2          __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::__normal_iterator(int const*
const&) [28]
-----------------------------------------------
                0.00    0.00       1/1          computeAverage(std::vector<int, std::allocator<int> > const&) [58]
[62]     0.0    0.00    0.00       1        std::vector<int, std::allocator<int> >::begin() const [62]
                0.00    0.00       1/2          __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::__normal_iterator(int const*
const&) [28]
-----------------------------------------------

 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 member of a cycle, the cycle number is printed between 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:

self This is the amount of time that was propagated directly from the child into the function.

children This is the amount of time that was propagated from the child's children to the function.

called 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 `/'.

name 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.

Index by function name

```
  [58] computeAverage(std::vector<int, std::allocator<int> > const&) [21] std::vector<int, std::allocator<int> >::operator[](unsigned long) const [26]
std::vector<int, std::allocator<int> >::begin()
  [59] generateRandomNumbers(std::vector<int, std::allocator<int> >&) [29] void std::_Destroy_aux<true>::__destroy<int*>(int*, int*) [44]
std::vector<int, std::allocator<int> >::vector(unsigned long, std::allocator<int> const&)
  [17] isEven(int)            [30] std::_Vector_base<int, std::allocator<int> >::_M_allocate(unsigned long) [45] std::vector<int, std::allocator<int>
>::~vector()
  [18] isPrime(int)           [31] std::_Vector_base<int, std::allocator<int> >::_Vector_impl::_Vector_impl(std::allocator<int> const&) [12]
std::vector<int, std::allocator<int> >::operator[](unsigned long)
  [22] factorial(int)         [32] std::_Vector_base<int, std::allocator<int> >::_Vector_impl::~_Vector_impl() [46] void std::_Construct<int>(int*)
  [60] prefixSum(std::vector<int, std::allocator<int> > const&) [33] std::_Vector_base<int, std::allocator<int> >::_M_deallocate(int*, unsigned long)
[47] int* std::__fill_n_a<int*, unsigned long, int>(int*, unsigned long, int const&, std::random_access_iterator_tag)
  [28] __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::__normal_iterator(int const* const&) [34]
std::_Vector_base<int, std::allocator<int> >::_M_create_storage(unsigned long) [48] int* std::__addressof<int>(int&)
  [19] __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::operator++() [35] std::_Vector_base<int, std::allocator<int>
>::_Vector_impl_data::_Vector_impl_data() [49] std::__size_to_integer(unsigned long)
  [23] __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::__normal_iterator(int* const&) [27] std::_Vector_base<int,
std::allocator<int> >::_M_get_Tp_allocator() [50] int* std::__uninitialized_default_n<int*, unsigned long>(int*, unsigned long)
  [10] __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::operator++() [36] std::_Vector_base<int, std::allocator<int>
>::_Vector_base(unsigned long, std::allocator<int> const&) [51] int* std::__uninitialized_default_n_a<int*, unsigned long, int>(int*, unsigned long,
std::allocator<int>&)
  [16] bool __gnu_cxx::operator!=<int const*, std::vector<int, std::allocator<int> > >(__gnu_cxx::__normal_iterator<int const*, std::vector<int,
std::allocator<int> > > const&, __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > > const&) [37] std::_Vector_base<int,
std::allocator<int> >::~_Vector_base() [52] unsigned long const& std::min<unsigned long>(unsigned long const&, unsigned long const&)
   [9] bool __gnu_cxx::operator!=<int*, std::vector<int, std::allocator<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&) [38]
std::__new_allocator<int>::deallocate(int*, unsigned long) [14] __gnu_cxx::__enable_if<std::__is_integer<int>::__value, double>::__type
std::sqrt<int>(int)
  [13] __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::base() const [39]
std::__new_allocator<int>::allocate(unsigned long, void const*) [53] int* std::fill_n<int*, unsigned long, int>(int*, unsigned long, int const&)
  [20] __gnu_cxx::__normal_iterator<int const*, std::vector<int, std::allocator<int> > >::operator*() const [24]
std::__new_allocator<int>::~__new_allocator() [54] void std::_Destroy<int*>(int*, int*)
   [8] __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::base() const [40] int*
std::__uninitialized_default_n_1<true>::__uninit_default_n<int*, unsigned long>(int*, unsigned long) [55] void std::__fill_a<int*, int>(int*, int*, int
const&)
  [11] __gnu_cxx::__normal_iterator<int*, std::vector<int, std::allocator<int> > >::operator*() const [41] std::vector<int, std::allocator<int>
>::_S_max_size(std::allocator<int> const&) [56] __gnu_cxx::__enable_if<std::__is_scalar<int>::__value, void>::__type std::__fill_a1<int*,
int>(int*, int*, int const&)
  [61] std::vector<int, std::allocator<int> >::end() const [42] std::vector<int, std::allocator<int> >::_S_check_init_len(unsigned long,
std::allocator<int> const&) [57] operator new(unsigned long, void*)
  [15] std::vector<int, std::allocator<int> >::size() const [43] std::vector<int, std::allocator<int> >::_M_default_initialize(unsigned long)
  [62] std::vector<int, std::allocator<int> >::begin() const [25] std::vector<int, std::allocator<int> >::end()
```