

G. H. Raison College Of Engineering And Management, Wagholi Pune
Assignment no :- 1 2021- 2022

Department	<u>CE [SUMMER 2022 (Online)]</u>		
Term / Section	<u>III/B</u>	Date Of submission	<u>27-09-2021</u>
Subject Name /Code	<u>Data structure and algorithms / UCSL201 / UCSP201</u>		
Roll No.	<u>SCOB77</u>	Name	<u>Pratham Rajkumar pittu</u>
Registration Number	<u>2020AC0E1100107</u>		

EXPERIMENT NO. 1

- # Aim - Consider a student database of Sycomy class (at least 10 records). Database contains different fields of every student, like Roll No, Name, and SGPA. (array of objects of class).
- (a) Design a roll call list, arrange list of students according to roll numbers in ascending order (Bubble Sort).
 - (b) Arrange List of Students alphabetically (use Insertion Sort).
 - (c) Arrange list of Students to Find out first 10 toppers from a class (use quick Sort).
 - (d) Searching Students according to SGPA. If more than ~~the~~ one student having same SGPA, then print list of all students having same SGPA.
 - (e) Search a particular student according to name using binary search without recursion.

Objective →

- 1) To Study The concepts of array structure.
- 2) Apply different Sorting techniques on array of structure (Bubble, Insertion and Quick Sort).
- 3) display the output for every pass.
- 4) Apply different Sorting techniques on array of structure (Linear Search, Binary Search) and display output for every pass.
- 5) calculate time complexity.

Theory -

- (1) Structure - Structure is collection of different data items in such a way that they can be referred as single unit.

► Creating a Structure

```
struct student  
{  
    int roll-no;  
    char name[15];  
    float Sgpa;  
};
```

Here, Struct keyword used to declare Structure. Student is name of Structure. In the example, three types of variables int char float.

- Declaring Structure variable - After declaring Structure, we can define Structure variables. For instance, variables are defined with Structure data type of automobile.

```
struct student s1, s2, s3;
```

Here,

variables defined by struct of student. Suppose, to declare array of 20 elements we write

```
struct student s[20];
```

This creates 20 sets of variables that are organised.

Indeed, to access a specific Structure,

index array name

If we want to print name of 3rd element we write,

```
cout << S[2].name;
```

► Referring Structure Members with Dot operation -

Given the struct Student and S₁ is a variable of type struct student. We can access the fields in following ways:-

S₁.roll-no;

S₁.name;

S₁.sgpa;

► pointer to Structure

We can declare a pointer variable as ~~struct~~ struct

struct student * pointer or

struct student * ptr;

find suppose if we have Structure variable declared as;

struct ~~struc~~ student S₁;

Then ptr can store address by;

ptr = &S₁

► Referring a structure member with (→) :-

using ptr we can access fields of S₁ as

ptr → name or (*ptr).name

► Array of Structure - To declare an array, we first define a structure and then declare array variables of that type.

Sorting :->

(1) Bubble Sort :-> Bubble sort is simplest sorting algo method that works by repeatedly swapping adjacent elements if they are in wrong (improper) order.

Its avg and worst-case complexity of Bubble Sort is $O(n^2)$, where 'n' is a natural a no. of item.

Working -

Step 1 - Starting with first element, compare current element with next element of array.

Step 2 - If current element is greater than next, swap them.

Step 3 - If current element is less than next, move to next element.

Step 4 - Repeat Step 1, till list is sorted.

#2) Quick Sort - Quick Sort is algorithm based on divide and conquer approach in which array is split into sub-arrays and these sub-arrays are recursively called sort element.

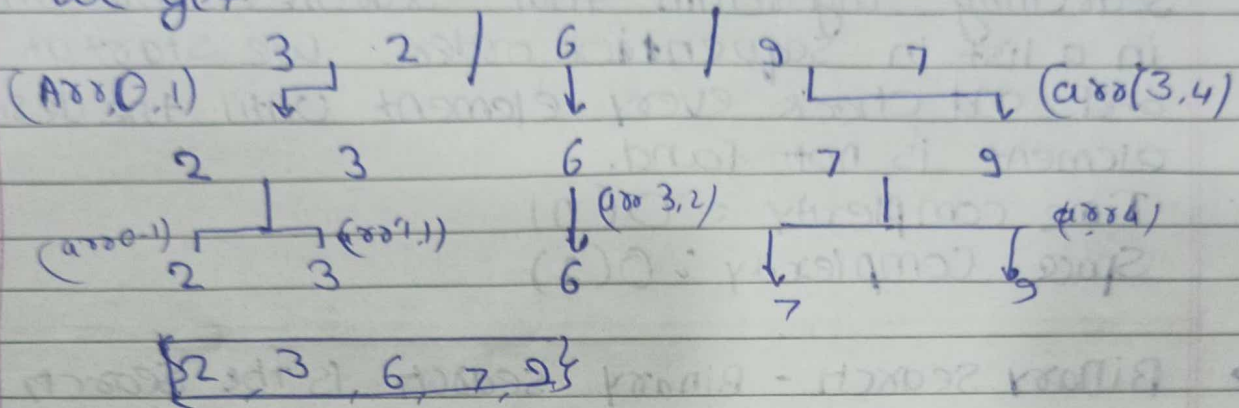
Its worst case complexity is $O(n^2)$. Avg case is $O(n \log n)$. It occurs when pivot element lies in extreme end of sorted array. One sub array is called also empty and another has $(n-1)$ elements.

example - consider an array as

$arr[5] = \{7, 9, 3, 2, 6\}$

0 1 2 3 4
7 9 3 2 6

we consider pivot elements as 6 (last element)
so and after we write $OS(arr, 0, 4)$
we get



3) Insertion Sort

Insertion Sort is simple sorting algorithm that works the way we sort playing cards in our hands. If we assume that the Insertion Sort first card is already sorted then, we select an unsorted card. If the unsorted card is greater than the card in hand, it is placed on the right otherwise, to the left.

Working:

- Step 1 \rightarrow pick next element
- Step 2 \rightarrow compare with all elements in sorted sublist.
- Step 3 \rightarrow Shift all elements in sorted sublist that are greater than value to be sorted on the left.
- Step 4 \rightarrow Insert value
- Step 5 \rightarrow Repeat until list is sorted.

Searching → It is a process of retrieving or locating a record with a particular key value.

► Linear Search →

Linear Search is the simplest searching algorithm that searches for an element in a list in sequential order. We start at one end and check every element until the desired element is not found.

Time complexity : $O(n)$

Space Complexity : $O(1)$

► Binary Search - Binary search is the search technique which works efficiently on the sorted lists. Hence, in order to search an element into some list by using binary search technique, we must ensure that the list is sorted.

Binary Search follows divide and conquer approach in which the list is divided into two halves and the item is compared with the middle element of the list. If the match is found then the location of middle element is returned otherwise we search into either of the halves depending upon the result produced through the match.

ALGORITHM / PSEUDOCODE :

➤ **Create a structure**

Create_database(struct student s[])

Step 1: Accept how many records user need to add, say, no of records as n

Step 2: For i = 0 to n – 1

i. Accept the record and store it in s[i]

Step 3: End For

Step 4: stop

Display_database(struct student s[] ,int n)

Step 1: For i = 0 to n –1

i. Display the fields s.roll_no, s.name, s.sgpa

Step 2: End For

Step 3: Stop

Bubble Sort student according to sort to roll numbers

BubbleSort(Student s[], n)

Step 1: For Pass = 1 to n-1

Step 2: For i = 0 to (n – pass – 1)

i. If s[i].roll_no < s[i+1].roll_no

20

a. Swap (s[i]. s[i+1])

iii. End if

Step 3: End for

Step 4: End For

Step 5 :Stop

➤ **Insertion Sort to sort student on the basis of names**

insertion_Sort (Struct student S[], int n)

Step 1: For i = 1 to n-1

i. Set key to s[i]

ii. Set j to i-1

iii. While j>=0 AND strcmp(s[i].name,key.name)>0

a. Assign s[j] to s[j+1]

b. Decrement j

iv. End While

Step 2: Assign key to s[j+1]

Step 3: End for

Step 4: end of insertion sort

➤ **Quick Sort to sort students on the basis of their sgpa.**

partition (struct student s[], int l, int h)

// where s is the array of structure , l is the index of starting element

// and h is the index of last element.

Step 1: Select s[l].sgpa as the pivot element

Step 2: Set i = l

Step 3: Set j = h-1

Step 4: While i ≤ j

i. Increment i till s[i].sgpa ≤ pivot element

ii. Decrement j till s[j].sgpa > pivot element

iii. If i < j

iv. Swap(s[i], s[j])

v. End if

Step 5:End while

Step 6: Swap(s[j],s[l])

Step 7: return j

Step 8 :end of Partition

quicksort(struct student s[], int l, int h)

//where s is the array of structure , l is the index of starting element

//and h is the index of last element.

Step 1: If l<h

i. P=partition(s,l,h)

ii. quicksort (s,l,p-1)

iii. quicksort (s,p+1,h)

Step 2: End if

Step 3: end of quicksort

21

➤ **Algorithm for Linear Search to search students with sgpa given and display all of them**

Linear_search (struct student s[], float key, int n)

//Here s is array of structure student, key is sgpa of student to be searched and

// displayed, n is total number of students in record

Step 1: Set i to 0 and flag to 0

Step 2: While i<n

i. If s[i].sgpa==key

a. Print s[i].roll_no, s[i].name

b. Set flag to 1

c. i++

Step 3: End while

Step 4: If flag==0

i. Print No student found with sgpa=value of key

Step 5: End if

Step 6: End of linear_sear

➤ **Algorithm for Binary Search to search students having given string in their names**

Binary_Search (s, n , Key)

// Where s is an array of structure , n is the no of records, and key is element to be searched

Step 1: Set l = 0 & h = n-1

Step 2:While l ≤ h

i. mid = (l + h) / 2

ii. If strcmp (s[mid].name, key)==0)

a. foun

b. stop

iii. Else

a. if (strcmp (key, s[mid].name)<0

i. h = mid – 1

b. Else

ii. l = mid + 1

c. End if

iv. End if

Step 4: End while

Step 5: not found // search is unsuccessful

Validation:

☐ Limit of the array should not be –ve, and should not cross the lower and upper bound.

☐ Roll numbers should not repeat, should not -ve

☐ Name should only contain alphabets, space and .

☐ Should not allowed any other operations before the input list is entered by the user .

☐ Before going to (binary search) records should be sorted according a names.

Test Cases :

☐ **Sorting :**

Four test cases :

22

i. Already Sorted according to the requirement

ii. Sorted in reverse order

iii. Partially sorted

iv. Completely Random List

Expected output /analysis is :

i. Test algorithm for above four test cases

ii. Analyze the algorithms based on no of comparisons and swapping/shifting required

iii. Check for Sort Stability factor

iv. No of passes needed

v. Best /average/ worst case of the each algorithm based on above test case

vi. Memory space required to sort

Searching :

☐ Find the max and minimum comparison required to search element

☐ Calculate how many comparisons are required for unsuccessful search

Application:

Useful in managing large amount of data.

Program code


```
#include <iostream>

#include <string.h>

using namespace std;

typedef struct student
{
    int roll_num;

    char name[20];

    float sgpa;
} stud;

void create(stud s[20], int n);

void display(stud s[20], int n);

void bubble_sort(stud s[20], int n);

void insertionSort(stud s[20], int n);

void quick_sort(stud s[20], int, int);

int partition(stud s[20], int, int);

void search(stud s[20], int n, float key);

int bsearch(stud s[20], char x[20], int low, int high);


int main()
{
    stud s[20];

    int ch, n, result;

    float key;

    char x[20];

    cout<<"\n\nSCOB77_pratham pittu\n\n";

    do
```

```

{
    cout << "\n 1) Create Student Database ";
    cout << "\n 2) Display Student Records ";
    cout << "\n 3) Bubble Sort ";
    cout << "\n 4) Insertion Sort ";
    cout << "\n 5) Quick Sort ";
    cout << "\n 6) Linear search ";
    cout << "\n 7) Binary search ";
    cout << "\n 8) Exit ";
    cout << "\n Enetr Your Choice:=";
    cin >> ch;
    switch (ch)
    {
    case 1:
        cout << "\n Enter The Number Of Records:=";
        cin >> n;
        create(s, n);
        break;
    case 2:
        display(s, n);
        break;
    case 3:
        bubble_sort(s, n);
        break;
    case 4:
        insertionSort(s, n);

```


break;

case 5:

quick_sort(s, 0, n - 1);

cout<<"\n"<< "\t"<< "Roll No"<< "\t\t"<< " Name" << "\t\t"<< "sgap";

for (int i = n - 1; i >= n - 10; i--)

{

cout << "\n";

cout << "\t " << s[i].roll_num << "\t\t " << s[i].name << "\t\t " << s[i].sgpa;

}

break;

case 6:

cout << "\n Enter the sgpa which u want to search:=";

cin >> key;

search(s, n, key);

break;

case 7:

cout << "\n Enter the name of student which u want to search:=";

cin >> x;

insertionSort(s, n);

result = bsearch(s, x, 0, (n - 1));

if (result == -1)

{

cout << " \n Student name you want to search for is not present ! \n";

}

else

{

```

        cout << " \n The student is present :\t" << s[result].name;
    }

    break;

case 8:

    return 0;

default:

    cout << "\n Invalid choice !! Please enter your choice again." << endl;

}

} while (ch != 8);

}

```

```

void create(stud s[20], int n)

```

```

{
    int i;

    for (i = 0; i < n; i++)

    {

        cout<<"\n Enter the ( roll number , Name , sgpa ) --> ";

        cin>>s[i].roll_num>>s[i].name>>s[i].sgpa;

    }

}

```

```

void display(stud s[20], int n)

```

```

{
    int i;

    cout<<"\n"<< "\t"<< "Roll No"<< "\t\t"<<" Name" <<"\t\t"<< "sgap";

    for (i = 0; i < n; i++)

    {

```



```

        cout << "\n";

        cout << "\t " << s[i].roll_num << "\t \t" << s[i].name << "\t \t" << s[i].sgpa;

    }

}

//bubble sort to sort in ascending order on roll number

```

```

void bubble_sort(stud s[20], int n)
{
    int i, j;

    stud temp;

    for (i = 1; i < n; i++)
    {
        for (j = 0; j < n - i; j++)
        {
            if (s[j].roll_num > s[j + 1].roll_num){
                temp = s[j];

                s[j] = s[j + 1];

                s[j + 1] = temp;
            }
        }
    }
}

```

// insertion sort to sort on names in ascending order

```

void insertionSort(stud s[20], int n)
{
    int i, j;

```

```

stud key;

for (i = 1; i < n; i++)
{
    key = s[i];

    j = i - 1;

    /* Move elements of arr[0..i-1], that are
    greater than key, to one position ahead
    of their current position */
    while (j >= 0 && strcmp(s[j].name, key.name) > 0)
    {
        s[j + 1] = s[j];

        j = j - 1;
    }
    s[j + 1] = key;
}
}

```

```

//Quick sort to sort on sgpa

void quick_sort(stud s[20], int l, int u)
{
    int j;

    if (l < u)
    {
        j = partition(s, l, u);

        quick_sort(s, l, j - 1);

        quick_sort(s, j + 1, u);
    }
}

```

```

    }
}

int partition(stud s[20], int l, int u)
{
    int i, j;
    stud temp, v;
    v = s[l];
    i = l;
    j = u + 1;
    do
    {
        do
            i++;

        while (s[i].sgpa < v.sgpa && i <= u);

        do
            j--;

        while (v.sgpa < s[j].sgpa);

        if (i < j)
        {
            temp = s[i];
            s[i] = s[j];
            s[j] = temp;
        }
    }
}

```



```
} while (i < j);
```

```
s[l] = s[j];
```

```
s[j] = v;
```

```
return (j);
```

```
}
```

```
// linear search for sgpa if more than one student having same sgpa print all of them
```

```
void search(stud s[20], int n, float key)
```

```
{
```

```
int i; int c=0;
```

```
cout<<"\n"<< "\t"<< "roll no"<< "\t\t"<< " name" << "\t\t"<< "sgpa";
```

```
for (i = 0; i < n; i++)
```

```
{
```

```
if (key == s[i].sgpa)
```

```
{
```

```
cout << "\n\t " << s[i].roll_num << "\t\t " << s[i].name << "\t\t " << s[i].sgpa;
```

```
c++;
```

```
}
```

```
}
```

```
if (c==0)
```

```
{
```

```
cout<<"No record found for this sgpa value";
```

```
}
```

```
}
```

```
int bsearch(stud s[20], char x[20], int low, int high)
{
    int mid;
    while (low <= high)
    {
        mid = (low + high) / 2;
        if (strcmp(x, s[mid].name) == 0)
        {
            return mid;
        }
        else if (strcmp(x, s[mid].name) < 0)
        {
            high = mid - 1;
        }
        else
        {
            low = mid + 1;
        }
    }
    return -1;
}
```

```
File Edit Selection View Go Run Terminal Help SCOB77_Pratham_pitty_DSA_assignment_1.cpp - vs code data - Visual Studio Code
SCOB77_Pratham_pitty_DSA_assignment_1.cpp X
SCOB77_Pratham_pitty_DSA_assignment_1.cpp > main()
15 int partition(stud s[100], int l, int h)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to
re-enable it, run 'Import-Module PSReadLine'.

PS C:\Users\prath\vs code data> cd "c:\Users\prath\vs code data\" ; if ($?) { g++ SCOB77_Pratham_pitty_DSA_assignment_1.cpp -o SCOB77_Prath
am_pitty_DSA_assignment_1 } ; if ($?) { .\SCOB77_Pratham_pitty_DSA_assignment_1 }

SCOB77_pratham_pitty

1) Create Student Database
2) Display Student Records
3) Bubble Sort
4) Insertion Sort
5) Quick Sort
6) Linear search
7) Binary search
8) Exit
Enetr Your Choice:=1

Enter The Number Of Records:=10

Enter the ( roll number , Name , sgpa ) --> 90 ram 9.8
Enter the ( roll number , Name , sgpa ) --> 100 shyam 9.4
Enter the ( roll number , Name , sgpa ) --> 80 jay 8.3
Enter the ( roll number , Name , sgpa ) --> 70 pratham 8.8
Enter the ( roll number , Name , sgpa ) --> 60 raju 8.9
Enter the ( roll number , Name , sgpa ) --> 50 ajay 7.2
Enter the ( roll number , Name , sgpa ) --> 40 sivam 6.6
```

```
File Edit Selection View Go Run Terminal Help SCOB77_Pratham_pitty_DSA_assignment_1.cpp - vs code data - Visual Studio Code
SCOB77_Pratham_pitty_DSA_assignment_1.cpp X
SCOB77_Pratham_pitty_DSA_assignment_1.cpp > main()
15 int partition(stud s[100], int l, int h)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Enter the ( roll number , Name , sgpa ) --> 40 sivam 6.6
Enter the ( roll number , Name , sgpa ) --> 30 akash 6.9
Enter the ( roll number , Name , sgpa ) --> 20 shashank 8.1
Enter the ( roll number , Name , sgpa ) --> 10 krisna 9.4

1) Create Student Database
2) Display Student Records
3) Bubble Sort
4) Insertion Sort
5) Quick Sort
6) Linear search
7) Binary search
8) Exit
Enetr Your Choice:=2

Roll No Name sgap
90 ram 9.8
100 shyam 9.4
80 jay 8.3
70 pratham 8.8
60 raju 8.9
50 ajay 7.2
40 sivam 6.6
30 akash 6.9
20 shashank 8.1
10 krisna 9.4

1) Create Student Database
2) Display Student Records
3) Bubble Sort
4) Insertion Sort
5) Quick Sort
6) Linear search
7) Binary search
8) Exit
Enetr Your Choice:=3

1) Create Student Database
2) Display Student Records
```



```
File Edit Selection View Go Run Terminal Help
SCOB77_Pratham_pitty_DSA_assignment_1.cpp - vs code data - Visual Studio Code

SCOB77_Pratham_pitty_DSA_assignment_1.cpp X
SCOB77_Pratham_pitty_DSA_assignment_1.cpp > main()
15 int partition(stud s[100], int l, int h)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1) Create Student Database
2) Display Student Records
3) Bubble Sort
4) Insertion Sort
5) Quick Sort
6) Linear search
7) Binary search
8) Exit
Enetr Your Choice:=2

Roll No      Name      sgap
10           krisna    9.4
20           shashank  8.1
30           akash     6.9
40           sivam     6.6
50           ajay      7.2
60           raju      8.9
70           pratham   8.8
80           jay       8.3
90           ram       9.8
100          shyam     9.4

1) Create Student Database
2) Display Student Records
3) Bubble Sort
4) Insertion Sort
5) Quick Sort
6) Linear search
7) Binary search
8) Exit
Enetr Your Choice:=4

1) Create Student Database
2) Display Student Records
3) Bubble Sort
4) Insertion Sort
5) Quick Sort
6) Linear search
7) Binary search
8) Exit
Enetr Your Choice:=2
```

```
SCOB77_Pratham_pitty_DSA_assignment_1.cpp X
SCOB77_Pratham_pitty_DSA_assignment_1.cpp > main()
15 int partition(stud s[100], int l, int h)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Enetr Your Choice:=2

Roll No      Name      sgap
50           ajay      7.2
30           akash    6.9
80           jay       8.3
10           krisna   9.4
70           pratham  8.8
60           raju      8.9
90           ram       9.8
20           shashank  8.1
100          shyam     9.4
40           sivam     6.6

1) Create Student Database
2) Display Student Records
3) Bubble Sort
4) Insertion Sort
5) Quick Sort
6) Linear search
7) Binary search
8) Exit
Enetr Your Choice:=5

Roll No      Name      sgap
90           ram       9.8
10           krisna   9.4
100          shyam     9.4
60           raju      8.9
70           pratham  8.8
80           jay       8.3
20           shashank  8.1
50           ajay      7.2
30           akash    6.9
40           sivam     6.6

1) Create Student Database
2) Display Student Records
3) Bubble Sort
4) Insertion Sort
5) Quick Sort
6) Linear search
7) Binary search
```

```
File Edit Selection View Go Run Terminal Help SCOB77_Pratham_pitty_DSA_assignment_1.cpp - vs code data - Visual Studio Code
SCOB77_Pratham_pitty_DSA_assignment_1.cpp X
SCOB77_Pratham_pitty_DSA_assignment_1.cpp > main()
15 int main() {
16     int n;
17     int arr[100];
18     int i;
19     int j;
20     int k;
21     int l;
22     int m;
23     int n;
24     int o;
25     int p;
26     int q;
27     int r;
28     int s;
29     int t;
30     int u;
31     int v;
32     int w;
33     int x;
34     int y;
35     int z;
36     int a;
37     int b;
38     int c;
39     int d;
40     int e;
41     int f;
42     int g;
43     int h;
44     int i;
45     int j;
46     int k;
47     int l;
48     int m;
49     int n;
50     int o;
51     int p;
52     int q;
53     int r;
54     int s;
55     int t;
56     int u;
57     int v;
58     int w;
59     int x;
60     int y;
61     int z;
62     int a;
63     int b;
64     int c;
65     int d;
66     int e;
67     int f;
68     int g;
69     int h;
70     int i;
71     int j;
72     int k;
73     int l;
74     int m;
75     int n;
76     int o;
77     int p;
78     int q;
79     int r;
80     int s;
81     int t;
82     int u;
83     int v;
84     int w;
85     int x;
86     int y;
87     int z;
88     int a;
89     int b;
90     int c;
91     int d;
92     int e;
93     int f;
94     int g;
95     int h;
96     int i;
97     int j;
98     int k;
99     int l;
100    int m;
101    int n;
102    int o;
103    int p;
104    int q;
105    int r;
106    int s;
107    int t;
108    int u;
109    int v;
110    int w;
111    int x;
112    int y;
113    int z;
114    int a;
115    int b;
116    int c;
117    int d;
118    int e;
119    int f;
120    int g;
121    int h;
122    int i;
123    int j;
124    int k;
125    int l;
126    int m;
127    int n;
128    int o;
129    int p;
130    int q;
131    int r;
132    int s;
133    int t;
134    int u;
135    int v;
136    int w;
137    int x;
138    int y;
139    int z;
140    int a;
141    int b;
142    int c;
143    int d;
144    int e;
145    int f;
146    int g;
147    int h;
148    int i;
149    int j;
150    int k;
151    int l;
152    int m;
153    int n;
154    int o;
155    int p;
156    int q;
157    int r;
158    int s;
159    int t;
160    int u;
161    int v;
162    int w;
163    int x;
164    int y;
165    int z;
166    int a;
167    int b;
168    int c;
169    int d;
170    int e;
171    int f;
172    int g;
173    int h;
174    int i;
175    int j;
176    int k;
177    int l;
178    int m;
179    int n;
180    int o;
181    int p;
182    int q;
183    int r;
184    int s;
185    int t;
186    int u;
187    int v;
188    int w;
189    int x;
190    int y;
191    int z;
192    int a;
193    int b;
194    int c;
195    int d;
196    int e;
197    int f;
198    int g;
199    int h;
200    int i;
201    int j;
202    int k;
203    int l;
204    int m;
205    int n;
206    int o;
207    int p;
208    int q;
209    int r;
210    int s;
211    int t;
212    int u;
213    int v;
214    int w;
215    int x;
216    int y;
217    int z;
218    int a;
219    int b;
220    int c;
221    int d;
222    int e;
223    int f;
224    int g;
225    int h;
226    int i;
227    int j;
228    int k;
229    int l;
230    int m;
231    int n;
232    int o;
233    int p;
234    int q;
235    int r;
236    int s;
237    int t;
238    int u;
239    int v;
240    int w;
241    int x;
242    int y;
243    int z;
244    int a;
245    int b;
246    int c;
247    int d;
248    int e;
249    int f;
250    int g;
251    int h;
252    int i;
253    int j;
254    int k;
255    int l;
256    int m;
257    int n;
258    int o;
259    int p;
260    int q;
261    int r;
262    int s;
263    int t;
264    int u;
265    int v;
266    int w;
267    int x;
268    int y;
269    int z;
270    int a;
271    int b;
272    int c;
273    int d;
274    int e;
275    int f;
276    int g;
277    int h;
278    int i;
279    int j;
280    int k;
281    int l;
282    int m;
283    int n;
284    int o;
285    int p;
286    int q;
287    int r;
288    int s;
289    int t;
290    int u;
291    int v;
292    int w;
293    int x;
294    int y;
295    int z;
296    int a;
297    int b;
298    int c;
299    int d;
300    int e;
301    int f;
302    int g;
303    int h;
304    int i;
305    int j;
306    int k;
307    int l;
308    int m;
309    int n;
310    int o;
311    int p;
312    int q;
313    int r;
314    int s;
315    int t;
316    int u;
317    int v;
318    int w;
319    int x;
320    int y;
321    int z;
322    int a;
323    int b;
324    int c;
325    int d;
326    int e;
327    int f;
328    int g;
329    int h;
330    int i;
331    int j;
332    int k;
333    int l;
334    int m;
335    int n;
336    int o;
337    int p;
338    int q;
339    int r;
340    int s;
341    int t;
342    int u;
343    int v;
344    int w;
345    int x;
346    int y;
347    int z;
348    int a;
349    int b;
350    int c;
351    int d;
352    int e;
353    int f;
354    int g;
355    int h;
356    int i;
357    int j;
358    int k;
359    int l;
360    int m;
361    int n;
362    int o;
363    int p;
364    int q;
365    int r;
366    int s;
367    int t;
368    int u;
369    int v;
370    int w;
371    int x;
372    int y;
373    int z;
374    int a;
375    int b;
376    int c;
377    int d;
378    int e;
379    int f;
380    int g;
381    int h;
382    int i;
383    int j;
384    int k;
385    int l;
386    int m;
387    int n;
388    int o;
389    int p;
390    int q;
391    int r;
392    int s;
393    int t;
394    int u;
395    int v;
396    int w;
397    int x;
398    int y;
399    int z;
400    int a;
401    int b;
402    int c;
403    int d;
404    int e;
405    int f;
406    int g;
407    int h;
408    int i;
409    int j;
410    int k;
411    int l;
412    int m;
413    int n;
414    int o;
415    int p;
416    int q;
417    int r;
418    int s;
419    int t;
420    int u;
421    int v;
422    int w;
423    int x;
424    int y;
425    int z;
426    int a;
427    int b;
428    int c;
429    int d;
430    int e;
431    int f;
432    int g;
433    int h;
434    int i;
435    int j;
436    int k;
437    int l;
438    int m;
439    int n;
440    int o;
441    int p;
442    int q;
443    int r;
444    int s;
445    int t;
446    int u;
447    int v;
448    int w;
449    int x;
450    int y;
451    int z;
452    int a;
453    int b;
454    int c;
455    int d;
456    int e;
457    int f;
458    int g;
459    int h;
460    int i;
461    int j;
462    int k;
463    int l;
464    int m;
465    int n;
466    int o;
467    int p;
468    int q;
469    int r;
470    int s;
471    int t;
472    int u;
473    int v;
474    int w;
475    int x;
476    int y;
477    int z;
478    int a;
479    int b;
480    int c;
481    int d;
482    int e;
483    int f;
484    int g;
485    int h;
486    int i;
487    int j;
488    int k;
489    int l;
490    int m;
491    int n;
492    int o;
493    int p;
494    int q;
495    int r;
496    int s;
497    int t;
498    int u;
499    int v;
500    int w;
501    int x;
502    int y;
503    int z;
504    int a;
505    int b;
506    int c;
507    int d;
508    int e;
509    int f;
510    int g;
511    int h;
512    int i;
513    int j;
514    int k;
515    int l;
516    int m;
517    int n;
518    int o;
519    int p;
520    int q;
521    int r;
522    int s;
523    int t;
524    int u;
525    int v;
526    int w;
527    int x;
528    int y;
529    int z;
530    int a;
531    int b;
532    int c;
533    int d;
534    int e;
535    int f;
536    int g;
537    int h;
538    int i;
539    int j;
540    int k;
541    int l;
542    int m;
543    int n;
544    int o;
545    int p;
546    int q;
547    int r;
548    int s;
549    int t;
550    int u;
551    int v;
552    int w;
553    int x;
554    int y;
555    int z;
556    int a;
557    int b;
558    int c;
559    int d;
560    int e;
561    int f;
562    int g;
563    int h;
564    int i;
565    int j;
566    int k;
567    int l;
568    int m;
569    int n;
570    int o;
571    int p;
572    int q;
573    int r;
574    int s;
575    int t;
576    int u;
577    int v;
578    int w;
579    int x;
580    int y;
581    int z;
582    int a;
583    int b;
584    int c;
585    int d;
586    int e;
587    int f;
588    int g;
589    int h;
590    int i;
591    int j;
592    int k;
593    int l;
594    int m;
595    int n;
596    int o;
597    int p;
598    int q;
599    int r;
600    int s;
601    int t;
602    int u;
603    int v;
604    int w;
605    int x;
606    int y;
607    int z;
608    int a;
609    int b;
610    int c;
611    int d;
612    int e;
613    int f;
614    int g;
615    int h;
616    int i;
617    int j;
618    int k;
619    int l;
620    int m;
621    int n;
622    int o;
623    int p;
624    int q;
625    int r;
626    int s;
627    int t;
628    int u;
629    int v;
630    int w;
631    int x;
632    int y;
633    int z;
634    int a;
635    int b;
636    int c;
637    int d;
638    int e;
639    int f;
640    int g;
641    int h;
642    int i;
643    int j;
644    int k;
645    int l;
646    int m;
647    int n;
648    int o;
649    int p;
650    int q;
651    int r;
652    int s;
653    int t;
654    int u;
655    int v;
656    int w;
657    int x;
658    int y;
659    int z;
660    int a;
661    int b;
662    int c;
663    int d;
664    int e;
665    int f;
666    int g;
667    int h;
668    int i;
669    int j;
670    int k;
671    int l;
672    int m;
673    int n;
674    int o;
675    int p;
676    int q;
677    int r;
678    int s;
679    int t;
680    int u;
681    int v;
682    int w;
683    int x;
684    int y;
685    int z;
686    int a;
687    int b;
688    int c;
689    int d;
690    int e;
691    int f;
692    int g;
693    int h;
694    int i;
695    int j;
696    int k;
697    int l;
698    int m;
699    int n;
700    int o;
701    int p;
702    int q;
703    int r;
704    int s;
705    int t;
706    int u;
707    int v;
708    int w;
709    int x;
710    int y;
711    int z;
712    int a;
713    int b;
714    int c;
715    int d;
716    int e;
717    int f;
718    int g;
719    int h;
720    int i;
721    int j;
722    int k;
723    int l;
724    int m;
725    int n;
726    int o;
727    int p;
728    int q;
729    int r;
730    int s;
731    int t;
732    int u;
733    int v;
734    int w;
735    int x;
736    int y;
737    int z;
738    int a;
739    int b;
740    int c;
741    int d;
742    int e;
743    int f;
744    int g;
745    int h;
746    int i;
747    int j;
748    int k;
749    int l;
750    int m;
751    int n;
752    int o;
753    int p;
754    int q;
755    int r;
756    int s;
757    int t;
758    int u;
759    int v;
760    int w;
761    int x;
762    int y;
763    int z;
764    int a;
765    int b;
766    int c;
767    int d;
768    int e;
769    int f;
770    int g;
771    int h;
772    int i;
773    int j;
774    int k;
775    int l;
776    int m;
777    int n;
778    int o;
779    int p;
780    int q;
781    int r;
782    int s;
783    int t;
784    int u;
785    int v;
786    int w;
787    int x;
788    int y;
789    int z;
790    int a;
791    int b;
792    int c;
793    int d;
794    int e;
795    int f;
796    int g;
797    int h;
798    int i;
799    int j;
800    int k;
801    int l;
802    int m;
803    int n;
804    int o;
805    int p;
806    int q;
807    int r;
808    int s;
809    int t;
810    int u;
811    int v;
812    int w;
813    int x;
814    int y;
815    int z;
816    int a;
817    int b;
818    int c;
819    int d;
820    int e;
821    int f;
822    int g;
823    int h;
824    int i;
825    int j;
826    int k;
827    int l;
828    int m;
829    int n;
830    int o;
831    int p;
832    int q;
833    int r;
834    int s;
835    int t;
836    int u;
837    int v;
838    int w;
839    int x;
840    int y;
841    int z;
842    int a;
843    int b;
844    int c;
845    int d;
846    int e;
847    int f;
848    int g;
849    int h;
850    int i;
851    int j;
852    int k;
853    int l;
854    int m;
855    int n;
856    int o;
857    int p;
858    int q;
859    int r;
860    int s;
861    int t;
862    int u;
863    int v;
864    int w;
865    int x;
866    int y;
867    int z;
868    int a;
869    int b;
870    int c;
871    int d;
872    int e;
873    int f;
874    int g;
875    int h;
876    int i;
877    int j;
878    int k;
879    int l;
880    int m;
881    int n;
882    int o;
883    int p;
884    int q;
885    int r;
886    int s;
887    int t;
888    int u;
889    int v;
890    int w;
891    int x;
892    int y;
893    int z;
894    int a;
895    int b;
896    int c;
897    int d;
898    int e;
899    int f;
900    int g;
901    int h;
902    int i;
903    int j;
904    int k;
905    int l;
906    int m;
907    int n;
908    int o;
909    int p;
910    int q;
911    int r;
912    int s;
913    int t;
914    int u;
915    int v;
916    int w;
917    int x;
918    int y;
919    int z;
920    int a;
921    int b;
922    int c;
923    int d;
924    int e;
925    int f;
926    int g;
927    int h;
928    int i;
929    int j;
930    int k;
931    int l;
932    int m;
933    int n;
934    int o;
935    int p;
936    int q;
937    int r;
938    int s;
939    int t;
940    int u;
941    int v;
942    int w;
943    int x;
944    int y;
945    int z;
946    int a;
947    int b;
948    int c;
949    int d;
950    int e;
951    int f;
952    int g;
953    int h;
954    int i;
955    int j;
956    int k;
957    int l;
958    int m;
959    int n;
960    int o;
961    int p;
962    int q;
963    int r;
964    int s;
965    int t;
966    int u;
967    int v;
968    int w;
969    int x;
970    int y;
971    int z;
972    int a;
973    int b;
974    int c;
975    int d;
976    int e;
977    int f;
978    int g;
979    int h;
980    int i;
981    int j;
982    int k;
983    int l;
984    int m;
985    int n;
986    int o;
987    int p;
988    int q;
989    int r;
990    int s;
991    int t;
992    int u;
993    int v;
994    int w;
995    int x;
996    int y;
997    int z;
998    int a;
999    int b;
1000   int c;
1001   int d;
1002   int e;
1003   int f;
1004   int g;
1005   int h;
1006   int i;
1007   int j;
1008   int k;
1009   int l;
1010   int m;
1011   int n;
1012   int o;
1013   int p;
1014   int q;
1015   int r;
1016   int s;
1017   int t;
1018   int u;
1019   int v;
1020   int w;
1021   int x;
1022   int y;
1023   int z;
1024   int a;
1025   int b;
1026   int c;
1027   int d;
1028   int e;
1029   int f;
1030   int g;
1031   int h;
1032   int i;
1033   int j;
1034   int k;
1035   int l;
1036   int m;
1037   int n;
1038   int o;
1039   int p;
1040   int q;
1041   int r;
1042   int s;
1043   int t;
1044   int u;
1045   int v;
1046   int w;
1047   int x;
1048   int y;
1049   int z;
1050   int a;
1051   int b;
1052   int c;
1053   int d;
1054   int e;
1055   int f;
1056   int g;
1057   int h;
1058   int i;
1059   int j;
1060   int k;
1061   int l;
1062   int m;
1063   int n;
1064   int o;
1065   int p;
1066   int q;
1067   int r;
1068   int s;
1069   int t;
1070   int u;
1071   int v;
1072   int w;
1073   int x;
1074   int y;
1075   int z;
1076   int a;
1077   int b;
1078   int c;
1079   int d;
1080   int e;
1081   int f;
1082   int g;
1083   int h;
1084   int i;
1085   int j;
1086   int k;
1087   int l;
1088   int m;
1089   int n;
1090   int o;
1091   int p;
1092   int q;
1093   int r;
1094   int s;
1095   int t;
1096   int u;
1097   int v;
1098   int w;
1099   int x;
1100   int y;
1101   int z;
1102   int a;
1103   int b;
1104   int c;
1105   int d;
1106   int e;
1107   int f;
1108   int g;
1109   int h;
1110   int i;
1111   int j;
1112   int k;
1113   int l;
1114   int m;
1115   int n;
1116   int o;
1117   int p;
1118   int q;
1119   int r;
1120   int s;
1121   int t;
1122   int u;
1123   int v;
1124   int w;
1125   int x;
1126   int y;
1127   int z;
1128   int a;
1129   int b;
1130   int c;
1131   int d;
1132   int e;
1133   int f;
1134   int g;
1135   int h;
1136   int i;
1137   int j;
1138   int k;
1139   int l;
1140   int m;
1141   int n;
1142   int o;
1143   int p;
1144   int q;
1145   int r;
1146   int s;
1147   int t;
1148   int u;
1149   int v;
1150   int w;
1151   int x;
1152   int y;
1153   int z;
1154   int a;
1155   int b;
1156   int c;
1157   int d;
1158   int e;
1159   int f;
1160   int g;
1161   int h;
1162   int i;
1163   int j;
1164   int k;
1165   int l;
1166   int m;
1167   int n;
1168   int o;
1169   int p;
1170   int q;
1171   int r;
1172   int s;
1173   int t;
1174   int u;
1175   int v;
1176   int w;
1177   int x;
1178   int y;
1179   int z;
1180   int a;
1181   int b;
1182   int c;
1183   int d;
1184   int e;
1185   int f;
1186   int g;
1187   int h;
1188   int i;
1189   int j;
1190   int k;
1191   int l;
1192   int m;
1193   int n;
1194   int o;
1195   int p;
1196   int q;
1197   int r;
1198   int s;
1199   int t;
1200   int u;
1201   int v;
1202   int w;
1203   int x;
1204   int y;
1205   int z;
1206   int a;
1207   int b;
1208   int c;
1209   int d;
1210   int e;
1211   int f;
1212   int g;
1213   int h;
1214   int i;
1215   int j;
1216   int k;
1217   int l;
1218   int m;
1219   int n;
1220   int o;
1221   int p;
1222   int q;
1223   int r;
1224   int s;
1225   int t;
1226   int u;
1227   int v;
1228   int w;
1229   int x;
1230   int y;
1231   int z;
1232   int a;
1233   int b;
1234   int c;
1235   int d;
1236   int e;
1237   int f;
1238   int g;
1239   int h;
1240   int i;
1241   int j;
1242   int k;
1243   int l;
1244   int m;
1245   int n;
1246   int o;
1247   int p;
1248   int q;
1249   int r;
1250   int s;
1251   int t;
1252   int u;
1253   int v;
1254   int w;
1255   int x;
1256   int y;
1257   int z;
1258   int a;
1259   int b;
1260   int c;
1261   int d;
1262   int e;
1263   int f;
1264   int g;
1265   int h;
1266   int i;
1267   int j;
1268   int k;
1269   int l;
1270   int m;
1271   int n;
1272   int o;
1273   int p;
1274   int q;
1275   int r;
1276   int s;
1277   int t;
1278   int u;
1279   int v;
1280   int w;
1281   int x;
1282   int y;
1283   int z;
1284   int a;
1285   int b;
1286   int c;
1287   int d;
1288   int e;
1289   int f;
1290   int g;
1291   int h;
1292   int i;
1293   int j;
1294   int k;
1295   int l;
1296   int m;
1297   int n;
1298   int o;
1299   int p;
1300   int q;
1301   int r;
1302   int s;
1303   int t;
1304   int u;
1305   int v;
1306   int w;
1307   int x;
1308   int y;
1309   int z;
1310   int a;
1311   int b;
1312   int c;
1313   int d;
1314   int e;
1315   int f;
1316   int g;
1317   int h;
1318   int i;
1319   int j;
1320   int k;
1321   int l;
1322   int m;
1323   int n;
1324   int o;
1325   int p;
1326   int q;
1327   int r;
1328   int s;
1329   int t;
1330   int u;
1331   int v;
1332   int w;
1333   int x;
1334   int y;
1335   int z;
1336   int a;
1337   int b;
1338   int c;
1339   int d;
1340   int e;
1341   int f;
1342   int g;
1343   int h;
1344   int i;
1345   int j;
1346   int k;
1347   int l;
1348   int m;
1349   int n;
1350   int o;
1351   int p;
1352   int q;
1353   int r;
1354   int s;
1355   int t;
1356   int u;
1357   int v;
1358   int w;
1359   int x;
1360   int y;
1361   int z;
1362   int a;
1363   int b;
1364   int c;
1365   int d;
1366   int e;
1367   int f;
1368   int g;
1369   int h;
1370   int i;
1371   int j;
1372   int k;
1373   int l;
1374   int m;
1375   int n;
1376   int o;
1377   int p;
1378   int q;
1379   int r;
1380   int s;
1381   int t;
1382   int u;
1383   int v;
1384   int w;
1385   int x;
1386   int y;
1387   int z;
1388   int a;
1389   int b;
1390   int c;
1391   int d;
1392   int e;
1393   int f;
1394   int g;
1395   int h;
1396   int i;
1397   int j;
1398   int k;
1399   int l;
1400   int m;
1401   int n;
1402   int o;
1403   int p;
1404   int q;
1405   int r;
1406   int s;
1407   int t;
1408   int u;
1409   int v;
1410   int w;
1411   int x;
1412   int y;
1413   int z;
1414   int a;
1415   int b;
1416   int c;
1417   int d;
1418   int e;
1419   int f;
1420   int g;
1421   int h;
1422   int i;
1423   int j;
1424   int k;
1425   int l;
1426   int m;
1427   int n;
1428   int o;
1429   int p;
1430   int q;
1431   int r;
1432   int s;
1433   int t;
1434   int u;
1435   int v;
1436   int w;
1437   int x;
1438   int y;
1439   int z;
1440   int a;
1441   int b;
1442   int c;
1443   int d;
1444   int e;
1445   int f;
1446   int g;
1447   int h;
1448   int i;
1449   int j;
1450   int k;
1451   int l;
1452   int m;
1453   int n;
1454   int o;
1455   int p;
1456   int q;
1457   int r;
1458   int s;
1459   int t;
1460   int u;
1461   int v;
1462   int w;
1463   int x;
1464   int y;
1465   int z;
1466   int a;
1467   int b;
1468   int c;
1469   int d;
1470   int e;
1471   int f;
1472   int g;
1473   int h;
1474   int i;
1475   int j;
1476   int k;
1477   int l;
1478   int m;
1479   int n;
1480   int o;
1481   int p;
1482   int q;
1483   int r;
1484   int s;
1485   int t;
1486   int u;
1487   int v;
1488   int w;
1489   int x;
1490   int y;
1491   int z;
1492   int a;
1493   int b;
1494   int c;
1495   int d;
1496   int e;
1497   int f;
1498   int g
```