

Lab 3 - Recursion

Write recursive functions for all the problems given below:

InLab

```
1. int sum(int x)
{ if (x<10) return x;
  else return x%10+sum(x/10);
}
main( )
{ int p;
  scanf("%d",&p);
  printf("%d\n",sum(p));
}
```

The above is a recursive function to find sum of digits. Similarly recursive function to delete k^{th} digit from last and maximum digit are defined.

```
int del(int x, int k)
{ if (k==1) return x/10;
  else return del(x/10,k-1)*10+x%10;
}
```

```
int max(int x)
{ int t;
  if (x<9) return x;
  else { t=max(x/10);
        if (t > x%10) return t;
        else return x%10;
      }
}
```

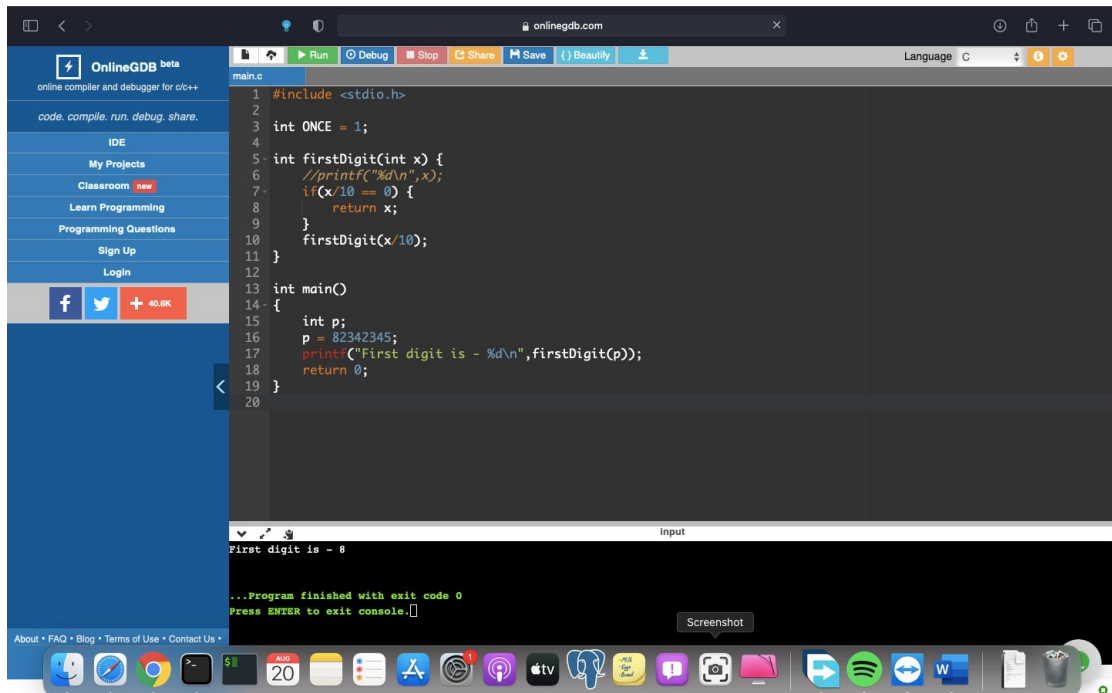
2) Define function to find first digit of a number.

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

```
int firstDigit(int x) {
    //printf("%d\n",x);
    if(x/10 == 0) {
        return x;
    }
    firstDigit(x/10);
}
```

```
int main()
{
    int p;
    p = 82342345;
    printf("First digit is - %d\n",firstDigit(p));
}
```

```
return 0;  
}
```



3) Define function to find second digit of a number.

```
#include <stdio.h>  
#include <setjmp.h>
```

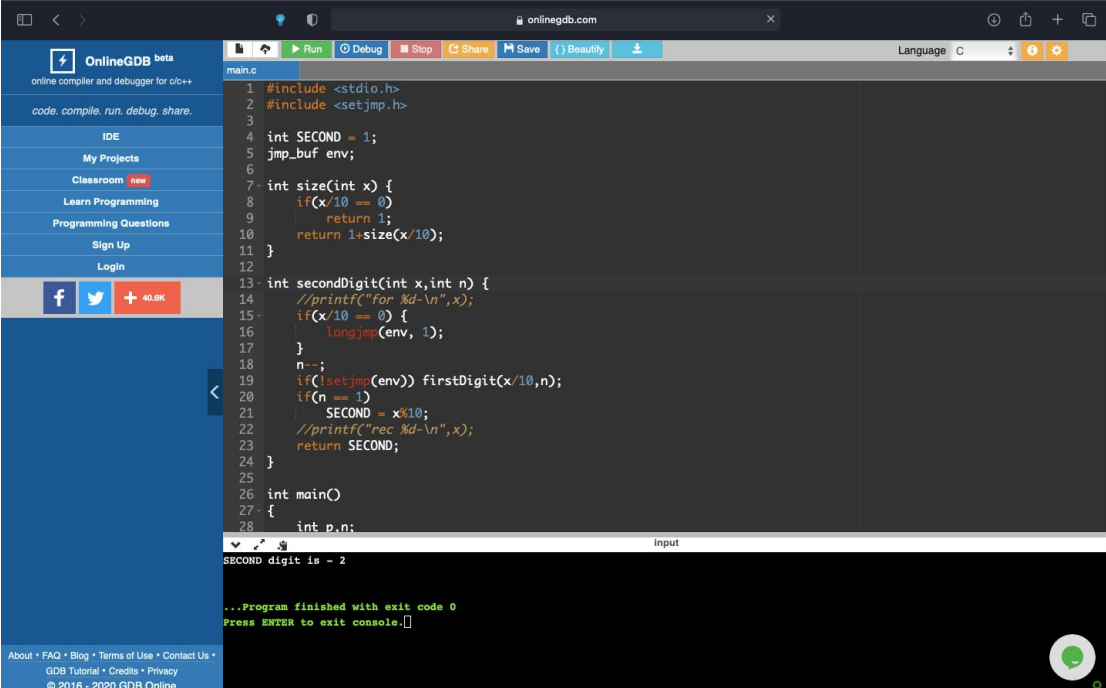
```
int SECOND = 1;  
jmp_buf env;
```

```
int size(int x) {  
    if(x/10 == 0)  
        return 1;  
    return 1+size(x/10);  
}
```

```
int firstDigit(int x,int n) {  
    //printf("for %d-\n",x);  
    if(x/10 == 0) {  
        return;  
    }  
    n--;  
    firstDigit(x/10,n);  
    if(n == 1)  
        SECOND = x%10;  
    //printf("rec %d-\n",x);
```

```
        return SECOND;
    }

    int main()
    {
        int p,n;
        p = 82342345;
        n = size(p);
        printf("First digit is - %d\n",firstDigit(p,n));
        return 0;
    }
```



The screenshot shows the OnlineGDB web interface. The code editor contains a C program with two functions: `size` and `secondDigit`. The `size` function recursively counts the number of digits in a number. The `secondDigit` function prints the first digit of a number and then calls `size` to find the second digit. The `main` function initializes `p` to 82342345 and calls `size` to find the number of digits. The output console shows the program finished with exit code 0 and the message "SECOND digit is - 2".

```
1 #include <stdio.h>
2 #include <setjmp.h>
3
4 int SECOND = 1;
5 jmp_buf env;
6
7 int size(int x) {
8     if(x/10 == 0)
9         return 1;
10    return 1+size(x/10);
11 }
12
13 int secondDigit(int x,int n) {
14     //printf("for %d\n",x);
15     if(x/10 == 0) {
16         longjmp(env, 1);
17     }
18     n--;
19     if(!setjmp(env)) firstDigit(x/10,n);
20     if(n == 1)
21         SECOND = x%10;
22     //printf("rec %d\n",x);
23     return SECOND;
24 }
25
26 int main()
27 {
28     int p,n;
```

SECOND digit is - 2

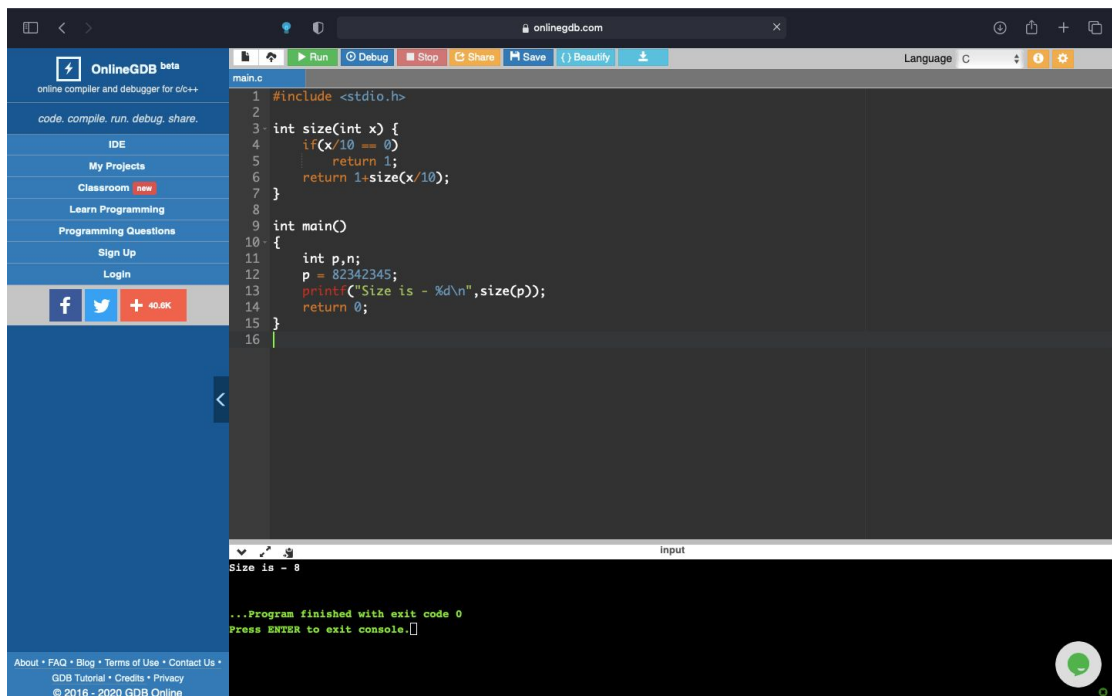
...Program finished with exit code 0
Press ENTER to exit console.

4) Define function to find the number of digits.

```
#include <stdio.h>

int size(int x) {
    if(x/10 == 0)
        return 1;
    return 1+size(x/10);
}

int main()
{
    int p,n;
    p = 82342345;
    printf("Size is - %d\n",size(p));
    return 0;
}
```



5) Take two arrays x and y. Put elements of array y in array x at the end.

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

```
int i = 1;
```

```
int n2 = 0; // as we can't change size dynamically. so defining globally
```

```
int* putAtEnd(int *arr1,int *arr2,int n1) {
    if(n1 == 0) {
        return 0;
    }
}
```

```
    putAtEnd(arr1,arr2,n1-1);
    arr2[n2] = arr1[n1-1];
    n2++;
}
```

```
void printIt(int *arr,int n) {
    for (int c = 0; c < n ; c++)
        printf("%d ", arr[c]);
}
```

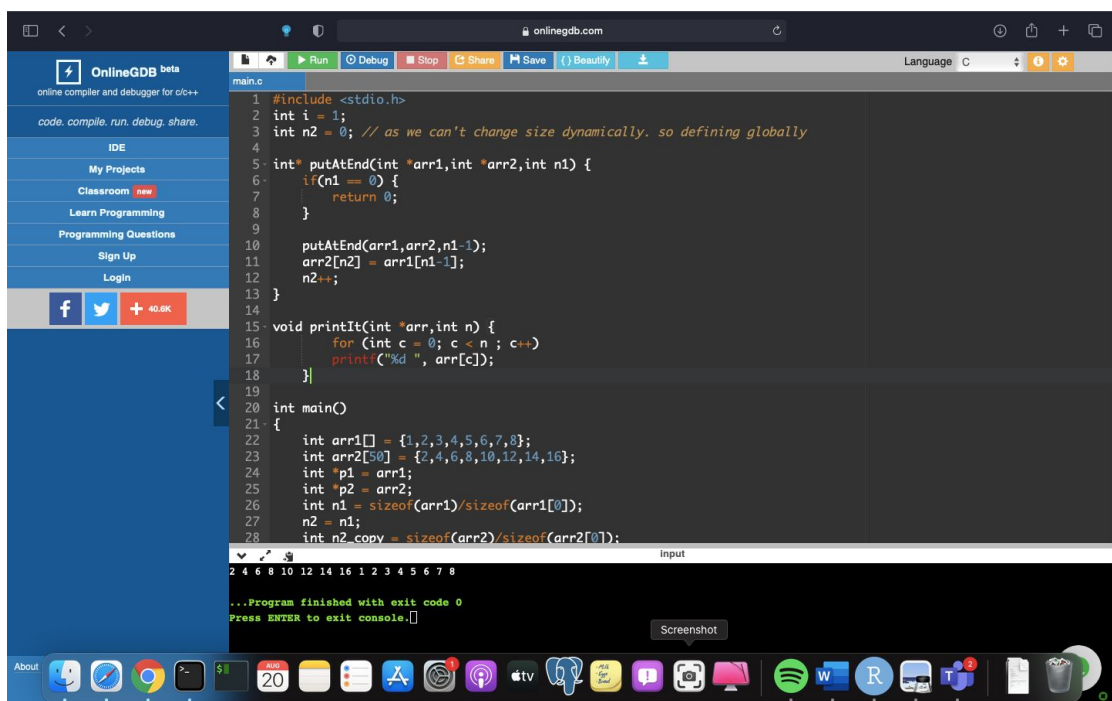
```
int main()
{
    int arr1[] = {1,2,3,4,5,6,7,8};
    int arr2[50] = {2,4,6,8,10,12,14,16};
}
```

```
int *p1 = arr1;
int *p2 = arr2;
int n1 = sizeof(arr1)/sizeof(arr1[0]);
n2 = n1;
int n2_copy = sizeof(arr2)/sizeof(arr2[0]);

putAtEnd(p1,p2,n1);
//int new2 = sizeof(arr2)/sizeof(arr2[0]);

printf(p2,n2);

return 0;
}
```

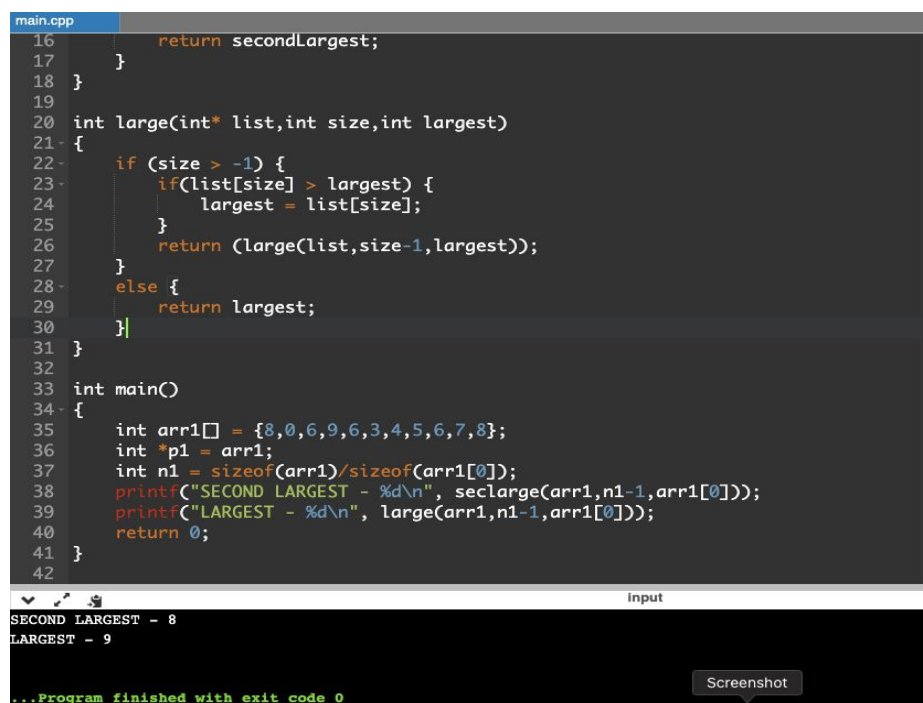


```
1 #include <stdio.h>
2 int i = 1;
3 int n2 = 0; // as we can't change size dynamically, so defining globally
4
5 int* putAtEnd(int *arr1,int *arr2,int n1) {
6     if(n1 == 0) {
7         return 0;
8     }
9
10    putAtEnd(arr1,arr2,n1-1);
11    arr2[n2] = arr1[n1-1];
12    n2++;
13 }
14
15 void printIt(int *arr,int n) {
16     for (int c = 0; c < n; c++)
17         printf("%d ", arr[c]);
18 }
19
20 int main()
21 {
22     int arr1[] = {1,2,3,4,5,6,7,8};
23     int arr2[50] = {2,4,6,8,10,12,14,16};
24     int p1 = arr1;
25     int p2 = arr2;
26     int n1 = sizeof(arr1)/sizeof(arr1[0]);
27     n2 = n1;
28     int n2_copy = sizeof(arr2)/sizeof(arr2[0]);
29
30     ...Program finished with exit code 0
31     Press ENTER to exit console.
```

6) Define function, which will return maximum and second maximum digit.

```
#include <stdio.h>
#define max(X, Y) (((X) >= (Y)) ? (X) : (Y))
int secondLargest;
int seclarge(int* list,int size,int largest)
{
    if (size > -1) {
        if(list[size] > largest) {
            secondLargest = largest;
            largest = list[size];
        }
    }
}
```

```
        return (seclarge(list,size-1,largest));
    }
    else {
        return secondLargest;
    }
}
int large(int* list,int size,int largest)
{
    if (size > -1) {
        if(list[size] > largest) {
            largest = list[size];
        }
        return (large(list,size-1,largest));
    }
    else {
        return largest;
    }
}
int main()
{
    int arr1[] = {8,0,6,9,6,3,4,5,6,7,8};
    int *p1 = arr1;
    int n1 = sizeof(arr1)/sizeof(arr1[0]);
    printf("SECOND LARGEST - %d\n", seclarge(arr1,n1-1,arr1[0]));
    printf("LARGEST - %d\n", large(arr1,n1-1,arr1[0]));
    return 0;
}
```



```
main.cpp
16         return secondLargest;
17     }
18 }
19
20 int large(int* list,int size,int largest)
21 {
22     if (size > -1) {
23         if(list[size] > largest) {
24             largest = list[size];
25         }
26         return (large(list,size-1,largest));
27     }
28     else {
29         return largest;
30     }
31 }
32
33 int main()
34 {
35     int arr1[] = {8,0,6,9,6,3,4,5,6,7,8};
36     int *p1 = arr1;
37     int n1 = sizeof(arr1)/sizeof(arr1[0]);
38     printf("SECOND LARGEST - %d\n", seclarge(arr1,n1-1,arr1[0]));
39     printf("LARGEST - %d\n", large(arr1,n1-1,arr1[0]));
40     return 0;
41 }
42
```

SECOND LARGEST - 8
LARGEST - 9

...Program finished with exit code 0

Screenshot

7) Define function to return maximum digit and how many times, it occurs. input 28688387 output 8 and 4.

```
#include<stdio.h>

int M_D(int a);
int i = 1;

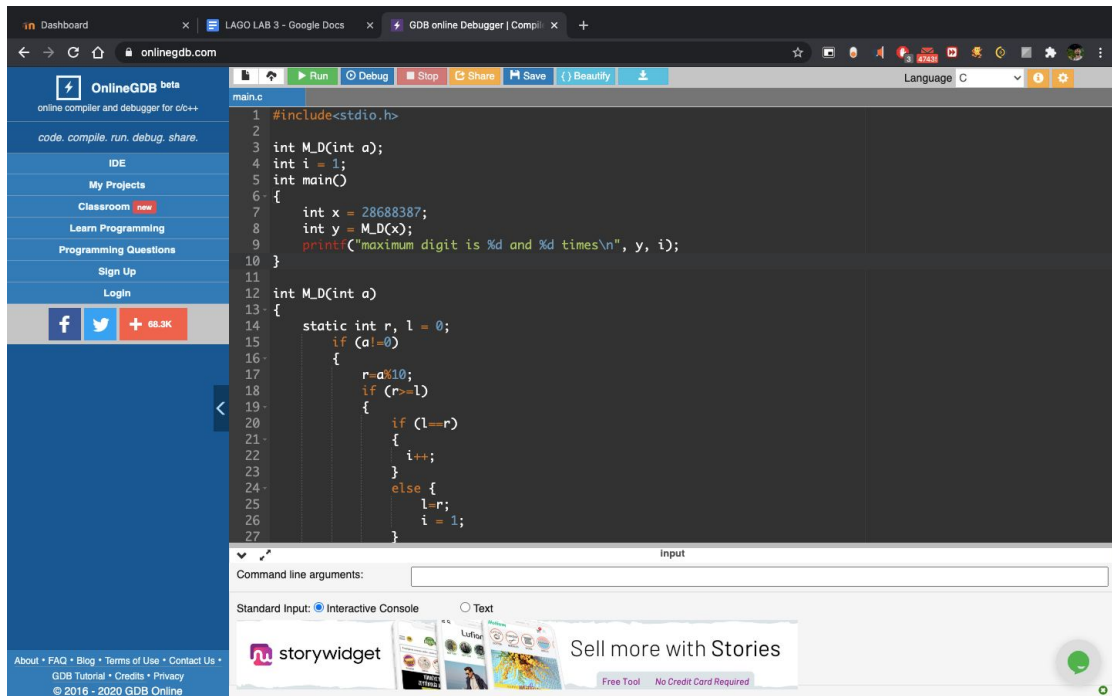
int main()
{
    int x = 28688387;
    int y = M_D(x);
    printf("maximum digit is %d and %d times\n", y, i);
}

int M_D(int a)
{
    static int r, l = 0;

    if (a!=0)
    {
        r=a%10;
        if (r>=l)
        {
            if (l==r)
            {
                i++;
            }
            else {
                l=r;
                i = 1;
            }
        }

        return (M_D(a/10));
    }

    return l;
}
```

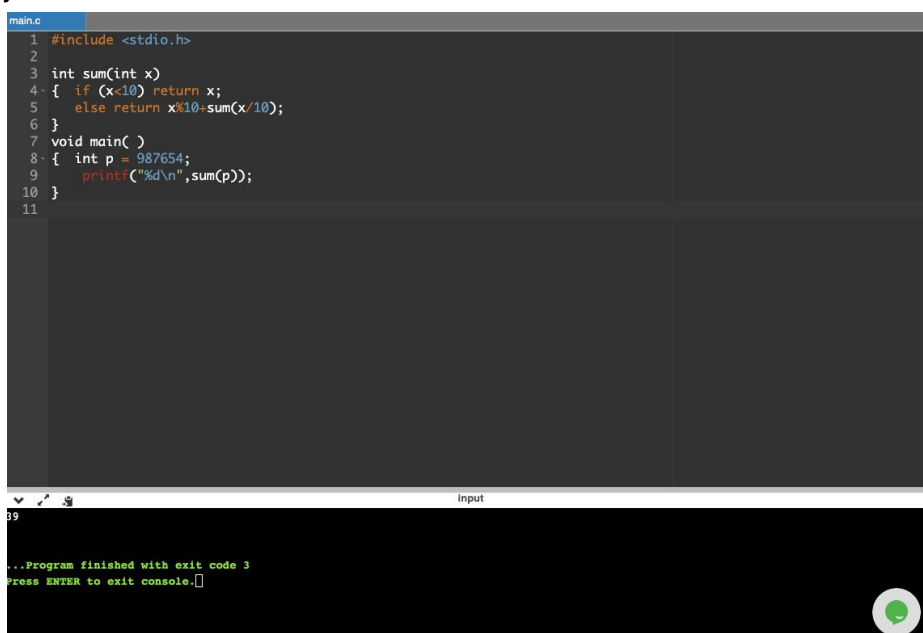


8) Write program to find the sum of all elements of the array.

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

```
int sum(int x)
{ if (x<10) return x;
  else return x%10+sum(x/10);
}
```

```
void main( )
{ int p = 987654;
  printf("%d\n",sum(p));
}
```



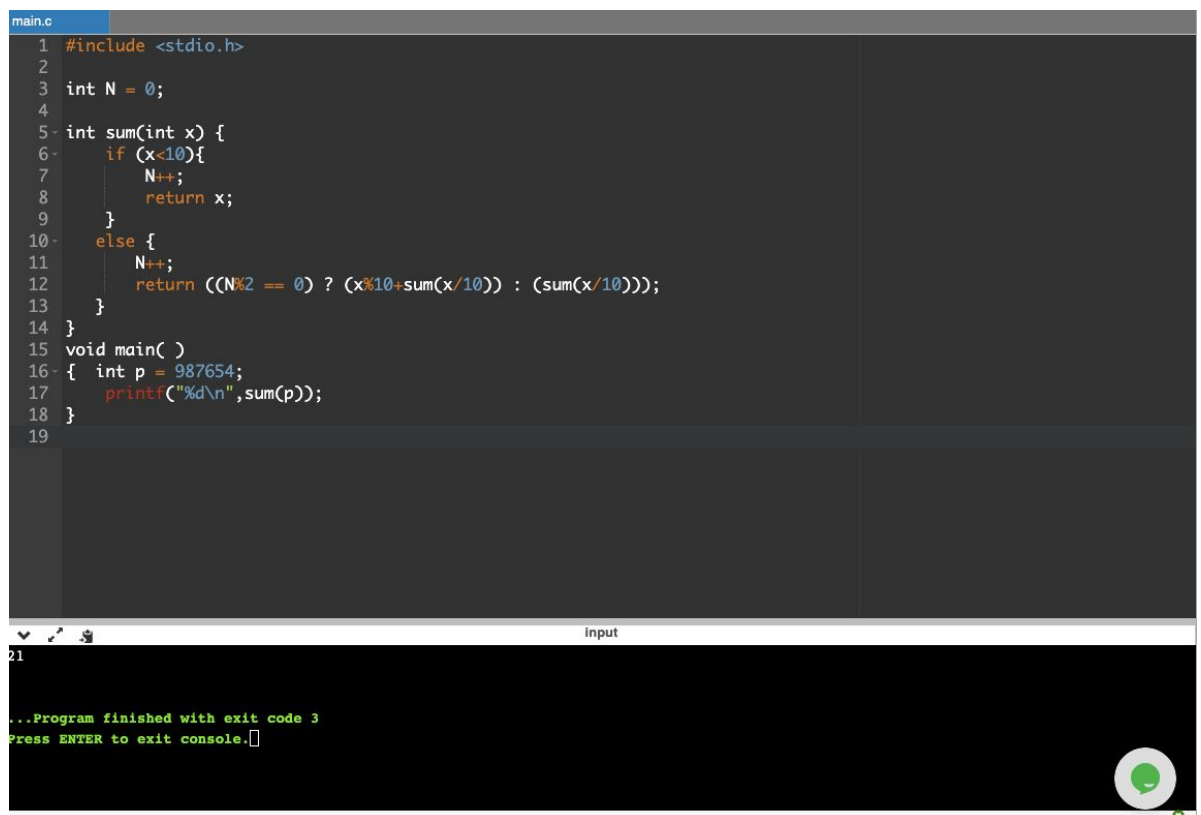
9) Write program to find the sum of all odd elements of the array.

```
#include <stdio.h>

int N = 0;

int sum(int x) {
    if (x<10) {
        N++;
        return x;
    }
    else {
        N++;
        return ((N%2 == 0) ? (x%10+sum(x/10)) : (sum(x/10)));
    }
}

void main( )
{
    int p = 987654;
    printf("%d\n",sum(p));
}
```

The image shows a screenshot of a C program being executed in a terminal window. The program is named 'main.c' and is located in the top-left corner of the window. The code is as follows:

```
1 #include <stdio.h>
2
3 int N = 0;
4
5 int sum(int x) {
6     if (x<10){
7         N++;
8         return x;
9     }
10    else {
11        N++;
12        return ((N%2 == 0) ? (x%10+sum(x/10)) : (sum(x/10)));
13    }
14 }
15 void main( )
16 { int p = 987654;
17   printf("%d\n",sum(p));
18 }
19
```

The program is executed, and the output is displayed in the terminal window. The output is '987654', which is the sum of all odd elements of the array [9, 8, 7, 6, 5, 4]. The terminal window also shows the message '...Program finished with exit code 3' and 'Press ENTER to exit console.'.

10) Define function, whose inputs are two numbers. The function returns two numbers. The first number is the number of digits of first number present in the second number. The second number is the number of digits of second number present in first number. input 2322677 and 13237 output 6 and 4. It is because digits 2, 3, 2, 2, 7 and 7 of first number are present in the second number.

```
#include <stdio.h>
#include <string.h>
int COUNT = 0;

char *toString(int num, char *str)
{
    if(str == NULL)
    {
        return NULL;
    }
    sprintf(str,"%d",num);
    return str;
}

char *toInt(char *str)
{
    if(str == NULL)
    {
        return NULL;
    }
    return (int)strtol(str, (char **)NULL, 10);
}

void quest10(int a,int b) {
    int bc = b;
    if (a<=0 || b<=0)
        return;
    while(bc>0) {
        if(a%10 == bc%10) {
            //printf("===%d===\n",COUNT);
            COUNT++;
        }
        bc/=10;
    }
    //printf("%d\t%d\n",a%10,b%10);
    quest10(a/10,b);
}
```

```
//printf("b = %d\n",b);
//quest10(a,b/10);
//(b>0 == 0) ? quest10(a,b/10) : quest10(a/10,b);
return;
}
```

```
void printIt(int *arr,int n) {
    for (int c = 0; c < n ; c++)
        printf("%d ", arr[c]);
}
```

```
void deleteduplicate(char *s,char c)
{
    int i,k=0;
    for(i=0;s[i];i++)
    {
        s[i]=s[i+k];
        if(s[i]==c)
        {
            k++;
            i--;
        }
    }
}
```

```
char findduplicate(char *s)
{
    char c='*';
    int i,j;

    for(i=0;s[i];i++)
    {
        if(!(s[i]==c))
        {
            for(j=i+1;s[j];j++)
            {
                if(s[i]==s[j])
                    s[j]=c;
            }
        }
    }
    return c;
}
```

```
int main()
{
    int an = 2322677;
    int bn = 13237;

    char aa[20];toString(an,aa);
    char ba[20];toString(bn,ba);
    char aad[20];toString(an,aad);
    char bad[20];toString(bn,bad);

    //int *aa = a;
    //int *bb = b;

    int n1 = strlen(aa);
    int n2 = strlen(ba);

    char rd1,rd2;
    rd1=findduplicate(aad);
    deleteduplicate(aad,rd1);

    rd2=findduplicate(bad);
    deleteduplicate(bad,rd2);

    int an_d = toInt(aad);
    int bn_d = toInt(bad);

    quest10(an,bn_d);
    printf("%d",COUNT);
    COUNT = 0;

    //printf("\n==%d==%d==",bn,an_d);

    quest10(bn,an_d);
    printf(" %d",COUNT);

    //printf("\n%d = %s",an,aad);
    return 69;

}
```

The screenshot displays the OnlineGDB web interface. The browser's address bar shows 'onlinegdb.com'. The left sidebar contains navigation links: 'code, compile, run, debug, share.', 'IDE', 'My Projects', 'Classroom' (with a 'new' badge), 'Learn Programming', 'Programming Questions', 'Sign Up', and 'Login'. Below these are social media icons for Facebook and Twitter, and a '+ 68.5K' button. The main editor area shows a C program with line numbers 93 to 119. The code includes comments, variable declarations, function calls, and a return statement. The bottom panel shows the compiler output with several warnings and a successful program completion message.

```
main.c
93 //int *bb = b;
94 int n1 = strlen(aa);
95 int n2 = strlen(ba);
96
97 char rd1,rd2;
98 rd1=findduplicate(aad);
99 deleteduplicate(aad,rd1);
100
101 rd2=findduplicate(bad);
102 deleteduplicate(bad,rd2);
103
104 int an_d = toInt(aad);
105 int bn_d = toInt(bad);
106
107 quest10(an,bn_d);
108 printf("%d",COUNT);
109 COUNT = 0;
110 //printf("\n==%d==%d==",bn,an_d);
111 quest10(bn,an_d);
112 printf("%d",COUNT);
113
114 //printf("\n%d = %s",an,aad);
115 return 69;
116
117
118 }
119
```

Input

```
main.c:21:17: warning: implicit declaration of function 'strlen' [-Wimplicit-function-declaration]
main.c:21:12: warning: return makes pointer from integer without a cast [-Wint-conversion]
main.c:104:16: warning: initialization makes integer from pointer without a cast [-Wint-conversion]
main.c:105:16: warning: initialization makes integer from pointer without a cast [-Wint-conversion]
6 4

...Program finished with exit code 69
Press ENTER to exit console.
```