WEEK-END ASSIGNMENT-10
Strings in C
Operating Systems Workshop (CSE 3541)

Problem Statement:

Experiment with character arrays, strings and operations on strings.

Assignment Objectives:

To understand how a string constant is stored in an array of characters. To learn about the placeholder %s and how it is used in **`printf`** and **`scanf`** operations. To learn operations performed on strings and also operations that can be performed on individual characters.

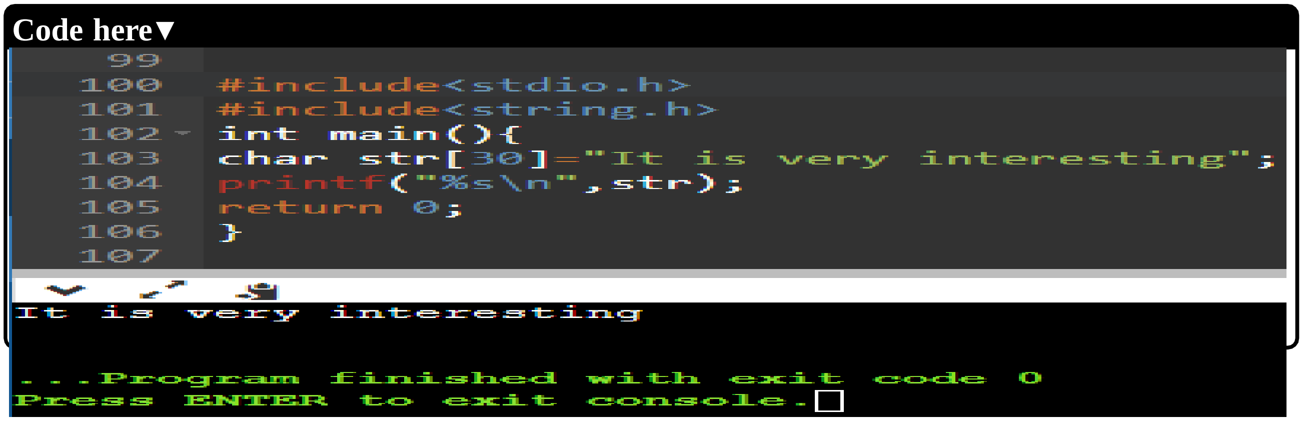
Instruction to Students (If any):

This assignment is designed to give practice with **strings, strings processing, and array of pointers in C**. Students are required to create their own programs to solve each and every question/problem as per the specification of the given question/problem to meet the basic requirement of systems programming. **Students are required to write the output/ paste the output screen shots onto their laboratory record after each question.**

Programming/ Output Based Questions:

1. We know a string in C is implemented as an array. So, declare and initialize the string **``It is very interesting''** and display the string.

Code here▼



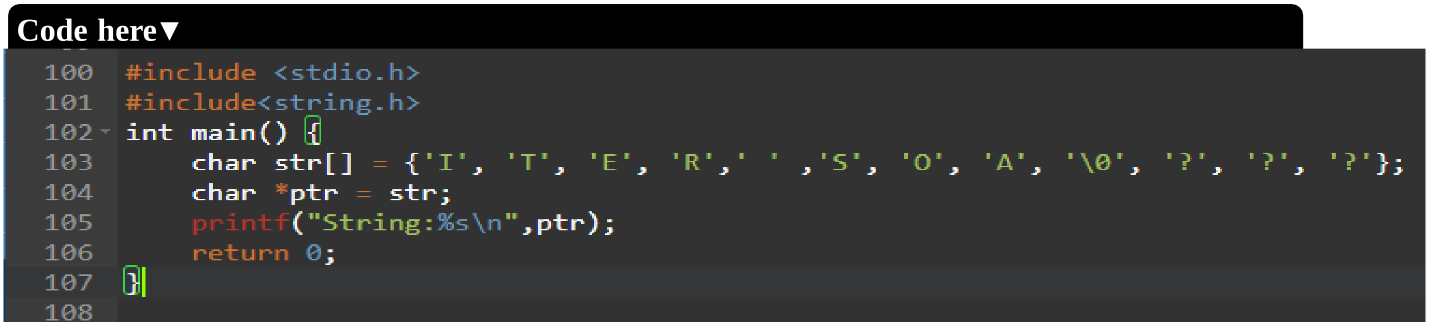
```
#include<stdio.h>
#include<string.h>
int main(){
char str[30]="It is very interesting";
printf("%s\n",str);
return 0;
}
```

```
It is very interesting

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

2. Declare and initialize the string using array reference and pointer.

0	1	2	3	4	5	6	7	8	9	10	11
I	T	E	R		S	O	A	\0	?	?	?

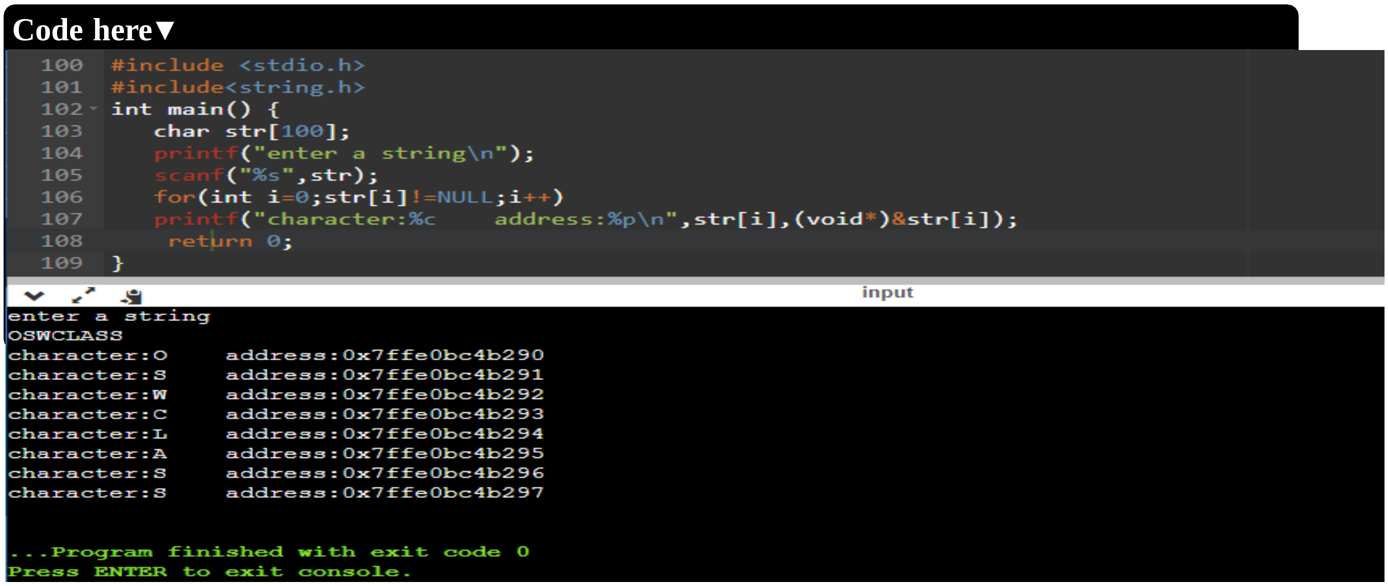
Code here▼



```
#include <stdio.h>
#include<string.h>
int main() {
    char str[] = {'I', 'T', 'E', 'R',' ' ,'S', 'O', 'A', '\0', '?', '?', '?'};
    char *ptr = str;
    printf("String:%s\n",ptr);
    return 0;
}
```

3. Write a program to read a string from the keyboard and print each character with their address on the screen.

Code here▼



```
#include <stdio.h>
#include<string.h>
int main() {
    char str[100];
    printf("enter a string\n");
    scanf("%s",str);
    for(int i=0;str[i]!=NULL;i++)
    printf("character:%c     address:%p\n",str[i],(void*)&str[i]);
     return 0;
}
```

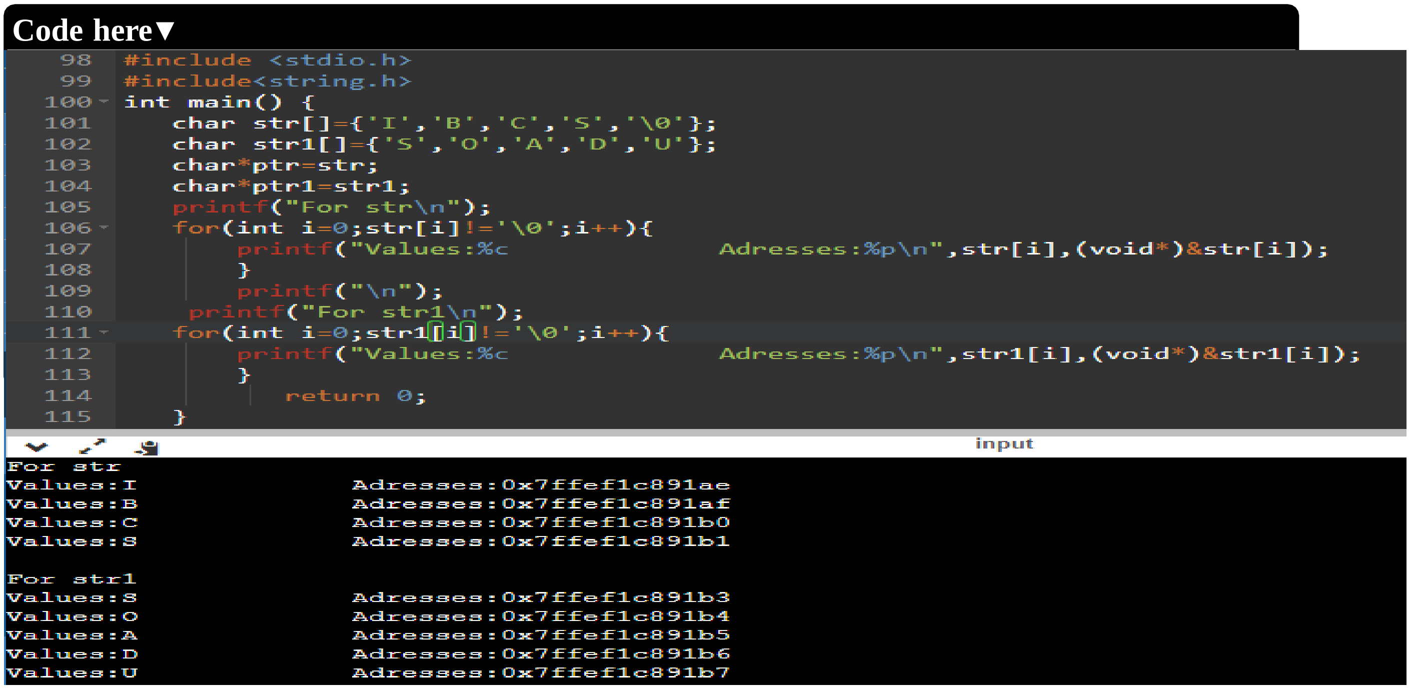
```
enter a string
OSWCLASS
character:O     address:0x7ffe0bc4b290
character:S     address:0x7ffe0bc4b291
character:W     address:0x7ffe0bc4b292
character:C     address:0x7ffe0bc4b293
character:L     address:0x7ffe0bc4b294
character:A     address:0x7ffe0bc4b295
character:S     address:0x7ffe0bc4b296
character:S     address:0x7ffe0bc4b297


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

4. Declare and initialize the two arrays to hold the values as shown in the given rectangular boxes. Writethe equivalent C statement to print their values and addresses using pointer.

0	1	2	3	4
I	B	C	S	\0

0	1	2	3	4
S	O	A	D	U

Code here▼

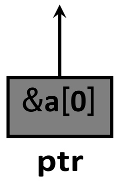


```
#include <stdio.h>
#include<string.h>
int main() {
    char str[]={'I','B','C','S','\0'};
    char str1[]={'S','O','A','D','U'};
    char*ptr=str;
    char*ptr1=str1;
    printf("For str\n");
    for(int i=0;str[i]!='\0';i++){
        printf("Values:%c           Adresses:%p\n",str[i],(void*)&str[i]);
        }
        printf("\n");
     printf("For str1\n");
    for(int i=0;str1[i]!='\0';i++){
        printf("Values:%c           Adresses:%p\n",str1[i],(void*)&str1[i]);
        }
            return 0;
    }
```

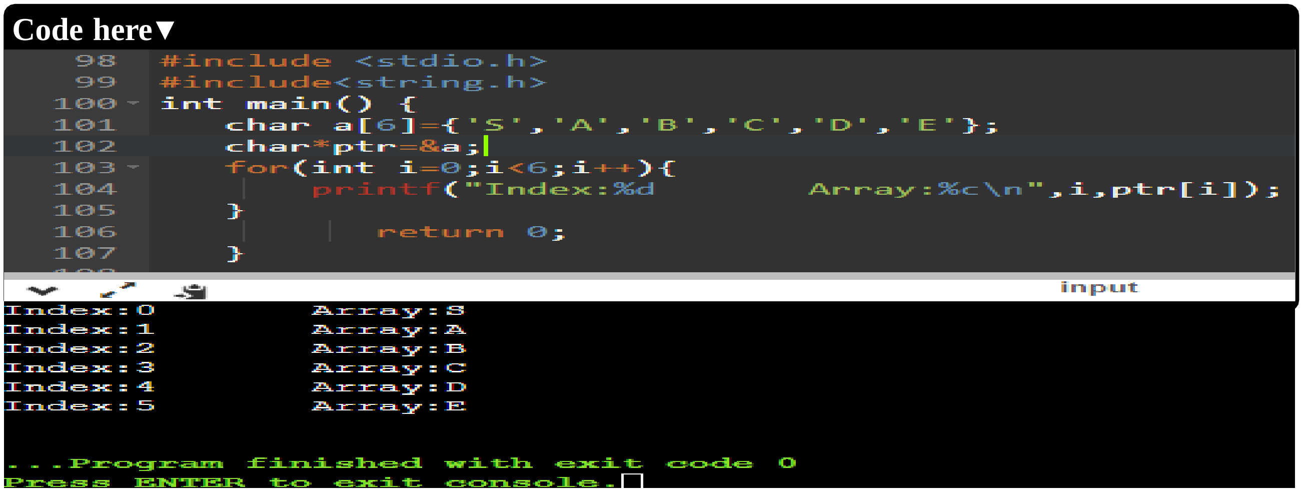
```
For str
Values:I             Adresses:0x7ffef1c891ae
Values:B             Adresses:0x7ffef1c891af
Values:C             Adresses:0x7ffef1c891b0
Values:S             Adresses:0x7ffef1c891b1

For str1
Values:S             Adresses:0x7ffef1c891b3
Values:O             Adresses:0x7ffef1c891b4
Values:A             Adresses:0x7ffef1c891b5
Values:D             Adresses:0x7ffef1c891b6
Values:U             Adresses:0x7ffef1c891b7
```

5. Write the C statement to declare and initialize the pointer variable for the given structure and displaythe array content using pointer.

Index	**0**	**1**	**2**	**3**	**4**	**5**
Array a	**S**	**A**	**B**	**C**	**D**	**E**



Code here▼



```
#include <stdio.h>
#include<string.h>
int main() {
      char a[6]={'S','A','B','C','D','E'};
      char*ptr=&a;
      for(int i=0;i<6;i++){
              printf("Index:%d          Array:%c\n",i,ptr[i]);
      }
                    return 0;
      }
```

```
Index:0           Array:S
Index:1           Array:A
Index:2           Array:B
Index:3           Array:C
Index:4           Array:D
Index:5           Array:E

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

6. For the given declarations `int  a[10]; int  pa;*` and assignment `pa=a;`, select the legal /illegal statements from the followings

```
(a) pa=a;     (f) is pa[i] identical to *(pa+i)?
(b) pa=&a[0]; (g) is &a[i] identical to (a+i)?
(c) pa++;     (h) is a[i] identical to  *(a+i)?
(d) a=pa      (i) is pa[i] identical to  a[i]?
(e) a++       (j) is void f(char str[]){...}identical to void f(char *str){...}?
(k) If a is an array, is f(&a[2]) identical to f(a+2);
```

Code here▼

LEGAL STATEMENTS:-

(a) pa=a;

(b) pa=&a[0];

(c) pa++;
(f) is pa[i] identical to *(pa+i) ?
(g) is &a[i] identical to (a+i) ?
(h) is a[i] identical to *(a+i) ?
(i) is pa[i] identical to a[i] ?
(j) is void f(char str[]){ . . . }identical to voidf(char*str){. . .} ?
(k) If a is an array, is f(&a[2]) identical to f(a+2);

ILLEGAL STATEMENTS:-

(d) a=pa
(e) a++

7. Let `p` be a pointer to an integer array and `n` is a scalar value. State the significance of the statement `p+n`.

Code here▼

In pointer arithmetic, adding an integer value n to a pointer p results in a new pointer that points to the memory location n elements away from the original location pointed to by p. The actual increment is based on the size of the data type to which the pointer points.
Here's a general formula:
***p*+*n*=address of *p*+(*n*×size of data type)**

8. If `arname` is an array, the function call `f(&arname[2]);` passes part of an array to the functionby passing a pointer to the begining of the sub-array. Write an equivalent statement for the call `f(&arname[2]);`.

Code here▼

The statement **f(&arname[2]);** passes the address of the third element of the array **arname** to the function **f**. This is equivalent to passing a pointer to the beginning of a sub-array starting from the third element
Another way to express this using pointer notation is:-**f(arname + 2);**
Both **&arname[2]** and **arname + 2** achieve the same result, passing a pointer to the third element of the array to the function.

9. Write an equivalent statement for the function's formal parameter **`a`**, whose header/definition is givenas **`f(int a[], int n, float y)`** {.........................}.

Code here▼

f(int *a, int n, float y) { ....}

10. Find the output of the following code segment for the function call **`bc=bytescount(``COVID-19 Still Active'')`**

```
int bytescount(char *s){
   char *p=s;
   while(*p!='\0'){
       p++;
   }
   return p-s;
}
```

Code here▼

bc=3

11. Find the output of the following code segment for the function call **`cc=countchar(``Encourgedto Vaccinate'')`**

```
int countchar(char *s){
   int n;
   for(n=0; *s!='\0'; s++){
       n++;
   }
   return n;
}
```

Code here▼

cc=23

12. Identify the type of variable **amsg** and **pmsg** from the following declaration and initialization statements.

```
char pmsg[]="I am in 5th Sem CSE";
char *amsg="I am in 5th Sem CSE";
```

Code here▼

pmsg is an array of characters.
amsg is a pointer to a character.

13. Find the output of the code fragment

```
char pmsg[60];
int nc;
nc=charcopy(pmsg,"I am in 5th Sem CSE");
printf("%d...%s\n",nc,pmsg);
```

The function definition/header is given as

```
int  charcopy(char *s, char *t)
{
   int                    i=0;
   while((s[i]=t[i])!='\0')
         i++;
   s[i]='\0';
   return(i);
}
```

Code here▼

19…I am in 5th Sem CSE

14. The function header is given as;

```
int  charcopy(char *s, char *t){
  int i=0;
  while((*s=*t)!=’\0’)
  {
        s++;
        t++;
        i++;
  }
  *s=’\0’;
  return(i);
}
```

Compute the output of the following code segment

```
char pmsg[60];
int nc;
nc=charcopy(pmsg,"Studied in CSE");
printf("%d...%s\n",nc,pmsg);
```

Code here▼

14…Studied in CSE

15. The function header is given as;

```
int  charcopy(char *s, char *t){
   int i=0;
   while((*s++=*t++)!=’\0’){
         i++;
   }
   *s=’\0’;
   return(i);
}
```

Compute the output of the following code segment

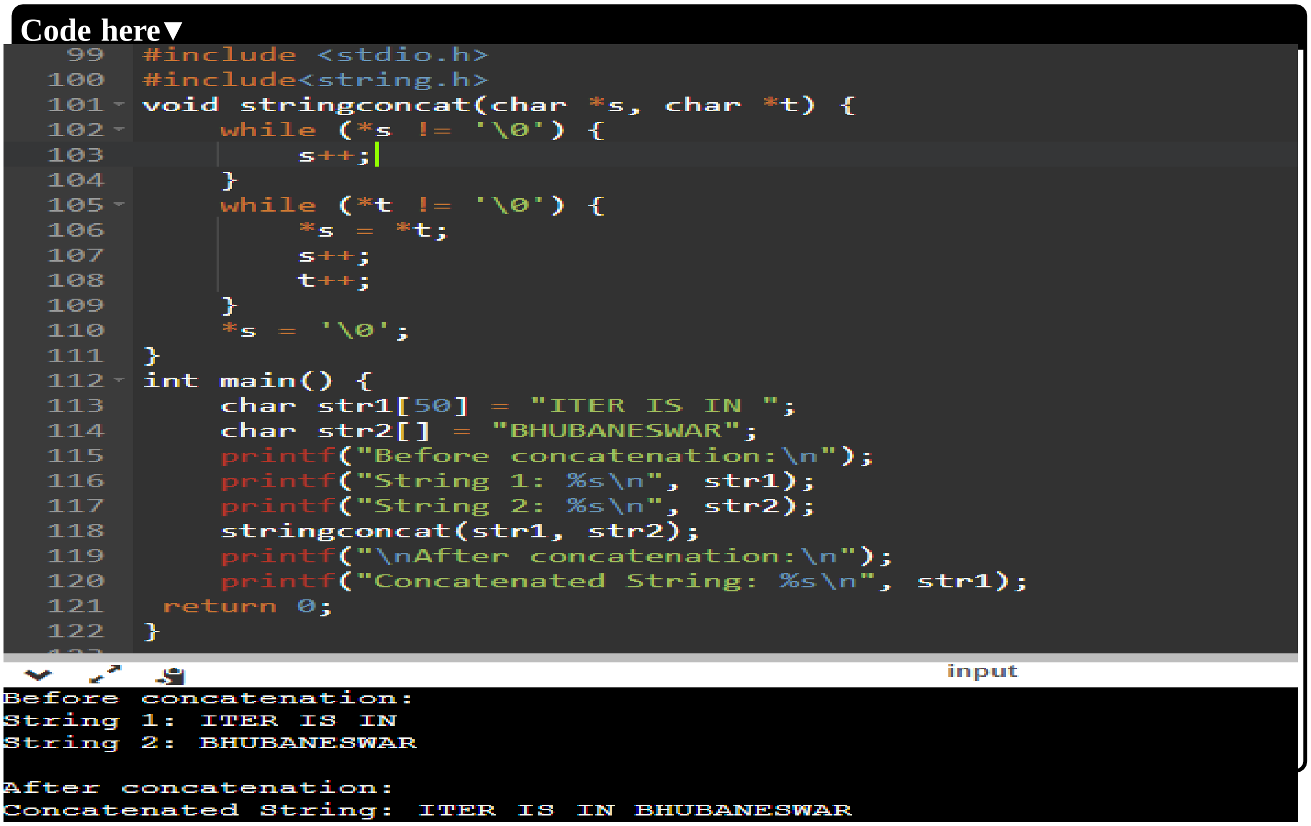
```
char pmsg[60];
int nc;
nc=charcopy(pmsg,"ITER CSE ");
printf("%d...%s\n",nc,pmsg);
```

Output here▼

9…ITER CSE

16. Write a pointer version of string concatenation program using the user-defined function, **`stringconcate(s,t);`**, copies the string **`t`** to the end **`s`**.

Code here▼



```
#include <stdio.h>
#include<string.h>
void stringconcat(char *s, char *t) {
    while (*s != '\0') {
        s++;
    }
    while (*t != '\0') {
        *s = *t;
        s++;
        t++;
    }
    *s = '\0';
}
int main() {
    char str1[50] = "ITER IS IN ";
    char str2[] = "BHUBANESWAR";
    printf("Before concatenation:\n");
    printf("String 1: %s\n", str1);
    printf("String 2: %s\n", str2);
    stringconcat(str1, str2);
    printf("\nAfter concatenation:\n");
    printf("Concatenated String: %s\n", str1);
  return 0;
}
```

```
Before concatenation:
String 1: ITER IS IN
String 2: BHUBANESWAR

After concatenation:
Concatenated String: ITER IS IN BHUBANESWAR
```

17. Write your own versions of the library functions **`strncpy`**, **`strncat`**, and **`strncmp`** which operate on at most the first **`n`** characters of their argument strings. For example **`strncpy(s,t,n)`** copiesat most **`n`** characters of **`t`** to **`s`**.

Code here▼

Code here▼

18. Write a program to take a product code from Millies Mail-Order Catalog (MMOC) and separate it into its component parts. An MMOC product code begins with one or more letters identifying the warehouse where the product is stored. Next come the one or more digits that are the product ID. The final field of the string starts with a capital letter and represents qualifiers such as size, color, and so on. For example, ATL1203S14 stands for product 1203, size 14, in the Atlanta warehouse. Write a program that takes a code, finds the position of the first digit and of the first letter after the digits, and uses **`strcpy`** and **`strncpy`** to display a report such as the following:

```
Warehouse: ATL
Product: 1203
Qualifiers: S14
```

Code here▼

Code here▼

19. Complete function **trim blanks(...)** whose purpose is to take a single string input parameter(**to trim**) and return a copy of the string with leading and trailing blanks removed. Use **strncpy**in **trim blanks**.

a_string (before)

		a		p	h	r	a	s	e					\0

n_string (after the call: trim_blanks(n_string, a_string);)

a		p	h	r	a	s	e	\0						

```
char * trim_blanks(char *trimmed,                  /* output */
                            const char *to_trim) /* input */
{
     /* Find subscript of first nonblank in to_trim */
     /* Find subscript of last nonblank in to_trim */
    /* Use strncpy to store trimmed string in trimmed */
}
```

Code here▼

20. Draw an array to show the output of the function, **`strcat(s1, s2)`**, used in the following code snippet and also write the output.

```
#define STRSIZ 20
char s1[STRSIZ]="Jupiter ", s2[STRSIZ]="Symphony";
printf("%d %d\n", strlen(s1), strlen(strcat(s1, s2)));
printf("%s\n", s1);
```

Code here▼

```
16 16
Jupiter Symphony
```

21. Given the string **`pres`** (value is **`"Adams, John Quincy"`**) and the 40-character temporary variables **`tmp1`** and **`tmp2`**, what string is displayed by the following code fragment?

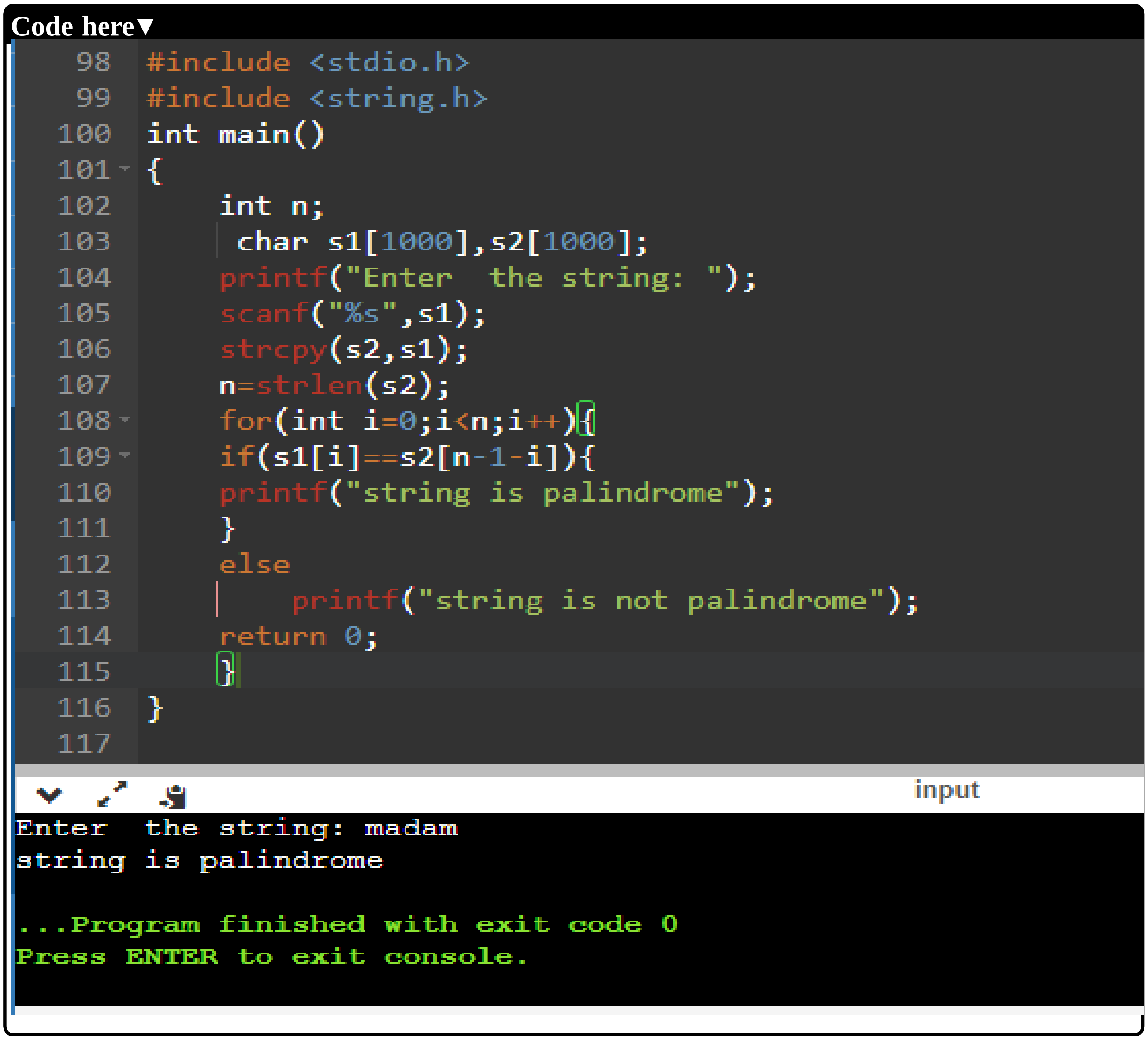
```
strncpy(tmp1, &pres[7], 4);
tmp1[4] = ’\0’;
strcat(tmp1, " ");
strncpy(tmp2, pres, 5);
tmp2[5] = ’\0’;
printf("%s\n", strcat(tmp1, tmp2));
```

Code here▼

```
John Adams
```

22. Write a program to check a string is palindrome or not. For example, **`madam`** is a palindrome, **`computer`** is not a palindrome.

Code here▼



```c
#include <stdio.h>
#include <string.h>
int main()
{
    int n;
     char s1[1000],s2[1000];
    printf("Enter  the string: ");
    scanf("%s",s1);
    strcpy(s2,s1);
    n=strlen(s2);
    for(int i=0;i<n;i++){
    if(s1[i]==s2[n-1-i]){
    printf("string is palindrome");
    }
    else
        printf("string is not palindrome");
    return 0;
    }
}
```

```
Enter  the string: madam
string is palindrome

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

23. Write a program in C to input a string using **`getchar()`** function only (Do not use **`scanf()`** or **`gets()`** function) and then count the total number of alphabets, number of alphabets in uppercase, number of alphabets in lowercase, number of digits, number of punctuation symbols, and number of spaces using character library functions.

Sample Run:

```
Input a string: I'm 2 bz 4 now.
Total number of alphabets: 7
Number of uppercase alphabets: 1
Number of lowercase alphabets: 6
Number of digits: 2
Number of punctuation mark: 2
Number of spaces: 4
```

Code here▼

```
#include <stdio.h>
#include <ctype.h>
int main() {
    char ch;
    int totalAlphabets = 0, uppercaseAlphabets = 0, lowercaseAlphabets = 0,
digits = 0, punctuationMarks = 0, spaces = 0;
printf("Input a string: ");
while ((ch = getchar()) != '\n') {
        if (isalpha(ch)) {
            totalAlphabets++;

            if (isupper(ch)) {
                uppercaseAlphabets++;
            } else {
                lowercaseAlphabets++;
            }
        } else if (isdigit(ch)) {
            digits++;
        } else if (ispunct(ch)) {
            punctuationMarks++;
        } else if (isspace(ch)) {
            spaces++;
        }
    }
    printf("Total number of alphabets: %d\n", totalAlphabets);
    printf("Number of uppercase alphabets: %d\n", uppercaseAlphabets);
    printf("Number of lowercase alphabets: %d\n", lowercaseAlphabets);
    printf("Number of digits: %d\n", digits);
    printf("Number of punctuation marks: %d\n", punctuationMarks);
    printf("Number of spaces: %d\n", spaces);
return 0;
}
```

24. Write a program in C to read N strings from user and then sort them using bubble sort.

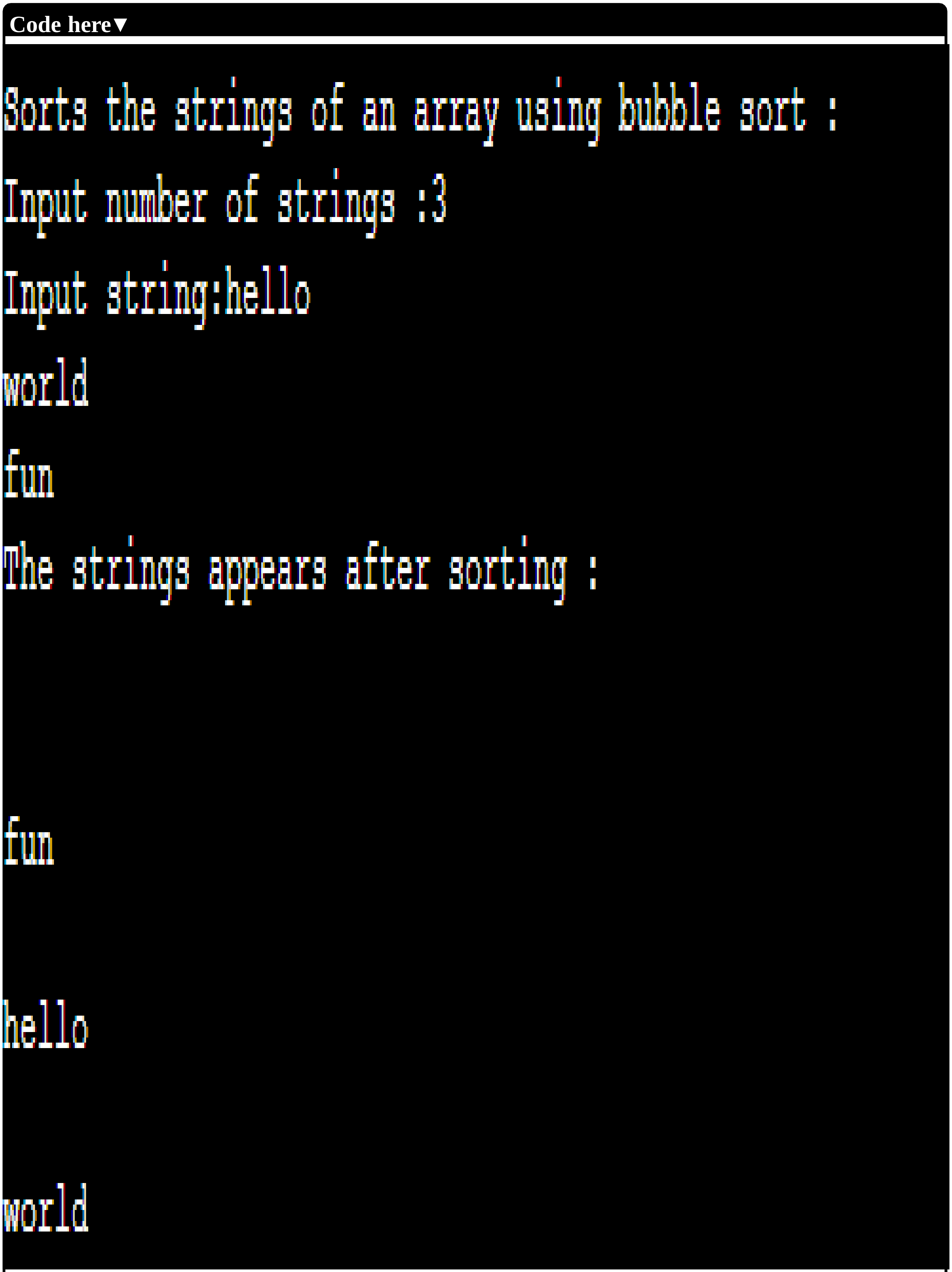
Sample Run:

```
Input number of strings :3
Input 3 strings:
hello
world
fun
The sorted strings are:
fun
hello
world
```

Code here▼

```
#include <stdio.h>
#include <string.h>
void main()
{
  char name[25][50],temp[25];
  int n,i,j;
  printf("\n\nSorts the strings of an array using bubble sort :\n");
  printf("Input number of strings :");
  scanf("%d",&n);
  printf("Input string:");
  for(i=0;i<=n;i++)
  {
    fgets(name[i], sizeof name, stdin);
  }
    for(i=1;i<=n;i++){
          for(j=0;j<=n-i;j++){
            if(strcmp(name[j],name[j+1])>0)
            {
            strcpy(temp,name[j]);
              strcpy(name[j],name[j+1]);
              strcpy(name[j+1],temp);
            }
          }
    }
   printf("The strings appears after sorting:\n");
for(i=0;i<=n;i++){
                     printf("%s\n",name[i]);
}
```

Code here▼



```
Sorts the strings of an array using bubble sort :
Input number of strings :3
Input string:hello
world
fun
The strings appears after sorting :



fun


hello


world
```

25. What is the value of **t1** after execution of these statementsif the value of **t2** is ‘‘**Merry Christmas**’’?

```
strncpy(t1, &t2[3], 5);
t1[5] = ’\0’;
```

Code here▼

```
ry ch
```

26. What does this program fragment display?

```
char x[80] = "gorilla";
char y[80] = "giraffe";
strcpy(x,  y);
printf("%s %s\n", x, y);
```

Code here▼

```
giraffe giraffe
```

27. What does this program fragment display?

```
char x[80] = "gorilla";
char y[80] = "giraffe";
strcat(x,  y);
printf("%s %s\n", x, y);
```

Code here▼

```
gorillagiraffe giraffe
```