

Square No.:

```
main()
{
    float a, x;
    printf("\n"); scanf("%f", &a);
    x = a * a;
    printf("%.f", x);
}
```

~~Square Root~~

~~#~~ <math.h>

- 1). ceil (number): rounds up given no. It returns int value.
- 2). floor (number): rounds down given no.
- 3). sqrt (number): returns the square root of a given no.
- 4). pow (base, exponent): returns power of no.
- 5). abs (number): returns the absolute value of no.

Note: Available in <math.h>

Sq. root:

```
#include <stdio.h>
#include <math.h>
main()
{
    float a, x;
    printf("\n"); scanf("%f", &a);
    x = sqrt(a);
    printf("%f", x);
}
```

Q. WAP C to print table of a no. (10).  
 (Ans):  

```

for (i=1; i<=10; i++)
    printf ("%d * %d = %d\n",
            n, i, n*i);
  
```

Note: gives  
 only `abs()` ~~take~~ and int value,  
`floor()` gives double value,  
 other func.'s give ~~double~~ value -  
 float

`Sqrt`  $\Rightarrow$  `%f`  
`pow()`  $\Rightarrow$  `%f`

SPX No.: (Both sum & product of factors (except itself)).

eg: 6 (✓)

Factors: 1, 2, 3. 6 (not included)

$$1 + 2 + 3 = 6$$

$$1 * 2 * 3 = 6$$

eg: 8 (✗)

Factors: 1, 2, 4  
 (✗)

$$1 + 2 + 4 = 7 \text{ (✗)}$$

$$1 * 2 * 4 = 8 \text{ (✓)}$$



## For single-Digit:

Page No.

Date / / 20

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

```
void main()
```

```
{ int n, i, fact-sum = 0, fact-prod = 1;
```

```
printf(" "); scanf("%d", &n);
```

```
for (i = 1; i < n; i++)
```

```
{ if (n % i == 0) // checking for factors
```

```
    fact-sum += i; // Adding factors
```

```
    fact-prod *= i; // Multiplying factors
```

```
}
```

```
}
```

```
// checking both conditions:
```

```
if ((fact-sum == n) && (fact-prod == n))
```

```
    printf("Spy Number\n");
```

```
else
```

```
    printf("Not a Spy Number\n");
```

```
}
```

## ~~For Multiple-Digit:~~

~~Sum of a given no. = Prod. of given no.  
Known as Spy No.~~

~~Sunny No.: (correct only)~~

~~I/P:~~

~~80 (✓)~~

~~$80 \times 1 = 81$~~

~~$\sqrt{81} = 9 \checkmark$~~

~~• Add 1 (✓)~~

~~• Sqrt() of a absolute no. (✓)~~

~~I/P:~~

~~7 (x)~~

~~$7 + 1 = 8$~~

~~$\sqrt{8} = ? (x)$~~



Spy-no. (Multiple - Digits):

```
while (num != 0)
{
    last dig = n % 10;
    sum = sum + dig;
    prod = prod * dig;
    n = n / 10;
}
```

Sunny No. :      Neon No. :

eg: 8      (x)  
      (2)  
      64      (x)  
      6 + 4 = 10

Sum of Sq. no's Digits == No.

```
main() (correct)
{
    a = 10;
    b = ++a + a++; 11 + 11 = 22 11 + 12 = 23
    printf(a, b);
}
```

```
a = 10;
b = ++a + a++;
printf(a, b);
```

O/P: (doubt)  
12, 23



$a++ + a++ \Rightarrow$  Dependency expression  
 $++a + ++a \Rightarrow$  Dependency expression

$++a + a++ \Rightarrow$  Indep.

$a++ + ++a \Rightarrow$  Indep.

$a = 10;$   
 $b = ++a + ++a + a++;$   
           13      13      10

Dependent

36

O/P:

13, 36

char array

Strings:

2 ways of rep:

→  $\text{char } a[] = \{ 'H', 'e', 'l', 'l', 'o', '\0' \}$

→  $\text{char } a[] = \text{"Hello"};$   
     ↳ string array

→ Here a → Address of first character is stored.

→ String is a sequence of characters.

→ It is enclosed within  $\{ \}$ .

→ It ends with  $\{ \}$ .

Syntax: datatype variable name [size]

[ ]

↳ without specifying also, we can declare strings.



(small s)

%s  $\Rightarrow$  string

We can also read and display output with help of ~~s~~ gets() and puts()

gets()  $\Rightarrow$  scanf()

puts()  $\Rightarrow$  printf()

#include <stdio.h>

#include <string.h>

void main()

{ char name[20];

printf("Enter string : ");

#scanf("%s", name);

#printf("Entered string\n");

}

gets(name);

puts(name);

String Functions: <string.h>

- 1) strlen (string\_name) : Length
- 2) strcpy (des., source) : copy of string
- 3) strcat (first, second) : concates / joins string
- 4) strcmp (first, second) : compares strings
- 5) strrev (string) : reverses string
- 6) strlwr (string) : returns upper case letters in str.
- 7) strupr (string) : returns lower case letters in str.



strcmp  $\Rightarrow 0 \Rightarrow$  same  
 $1 > 2 \Rightarrow 1$   
 $1 < 2 \Rightarrow -1$

Vowels:

main()

~~for~~ { char a[20];

int i, count = 0, cons = 0;

printf("Enter string: \n");

scanf("%s", &a);

strupr(a);

for (i = 0; i < strlen(a); i++)

{ if (a[i] == 'A' || a[i] == 'E' || a[i] == 'I' || a[i] == 'O' || a[i] == 'U')

count ++;

else

cons ++;

}

printf("Number of vowels: %d \n", count);

printf("Number of consonants: %d \n", cons);

}

## Frequency of char:

```
main()
{
    str[30];
    char
    char letter;

    int count = 0;
    i;

    printf("Enter String: \n");
    scanf("%s", &str);

    printf("Enter character: \n");
    scanf("%c", &letter);

    for (i = 0; str[i] != '\0'; i++)
    {
        if (str[i] == letter)
            count++;
    }
}
```

```
printf("%d", count);
```

## ~~Only letters from string~~ Duplicate char:

```
#include <stdio.h>
void main()
{
    char str[20];
    int
    printf(""); gets(str);
}
```



```

for (i=0; i<strlen(str); i++)
{
    for (j=i+1; j<strlen(str); j++)
    {
        if (str[i] == str[j])
            printf ("Duplicate: %c\n", str[i]);
    }
}
}

```

### Swapping Strings:

```

main()
{
    char a[20], b[20];
    int
    printf (""); gets (a);
    printf (""); gets (b);

    for (i=0; i<20; i++)
        temp[i] = a[i];

    for (i=0; i<20; i++)
        a[i] = b[i];

    for (i=0; i<20; i++)
        b[i] = temp[i];

    printf ("In str a: "); puts (a);
    printf ("In str b: "); puts (b);
}

```

HW:

Q1) Sorting strings

Q2) Removing spl. char & display only letters.



Array:  $(-1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$

- It is a collection of similar data items. It is stored in a sequential order.

→ Time:  $O(n)$  , Space:  $O(n)$

→ It is fixed in size.

→ Same Data Type (Homogeneous)

→ Accessed using index or subscript.

Dec:

```
datatype variable[size];  
name
```

### Initialising: