- String heading functions are defined under “string.h”header file.

#include<string.h>

strlen( )

```
string = Bamboozled length = 10
string = Humpty Dumpty length = 13
```

```
#include<stdio.h>
#include<string.h>
main()
{
  char arr[ ] = "Bamboozled" ;
  int len1, len2 ;
  len1 = strlen ( arr ) ;
  len2 = strlen ( "Humpty Dumpty" ) ;
  printf ( "\nstring = %s length = %d", arr, len1 ) ;
  printf ( "\nstring = %s length = %d", "Humpty Dumpty", len2 ) ;
}
```

strlen()

```
#include<stdio.h>
#include<string.h>
int main()
{
    char name[30];
    int x;
    printf("Enter name: ");
    gets(name);     //Function to read string from
user.
    x = strlen(name);
    printf("length =%d",x);
    return 0;
}
```

```
Enter name: Subhash Chandra Bose
length =20
--------------------------------
```

strcpy( )

- This function copies the contents of one string into another.
- The base addresses of the source and target strings should be supplied to this function.
- strcpy (target, source) ;

strcpy()

```
#include<stdio.h>
#include<string.h>
int main()
{
char source[ ] = "Sayonara" ;
char target[20] ;
strcpy ( target, source ) ;
printf ( "\nsource string = %s", source ) ;
printf ( "\ntarget string = %s", target ) ;
}
```

And here is the output...

source string = Sayonara
target string = Sayonara

strcat( )

- This function concatenates the source string at the end of the target string.
- strcat (target, source) ;

strcat( )

```
#include<stdio.h>
#include<string.h>
int main()
main( )
{
  char source[ ] = "Folks!" ;
  char target[30] = "Hello" ;
  strcat ( target, source ) ;
  printf ( "\nsource string = %s", source ) ;
  printf ( "\ntarget string = %s", target ) ;
}
```

And here is the output...

```
source string = Folks!
target string = HelloFolks!
```

strcmp( )

- This is a function which compares two strings to find out whether they are same or different.
- The two strings are compared character by character until there is a mismatch or end of one of the strings is reached, whichever occurs first.
- If the two strings are identical, **strcmp( )** returns a value zero.
- If they’re not, it returns the numeric difference between the ASCII values of the first non-matching pairs of characters.
- strcmp (string1, string2) ;

strcmp( )

```
#include<stdio.h>
#include<string.h>
int main()
{
   char string1[ ] = "Jerry" ;
   char string2[ ] = "Ferry" ;
   int i, j, k ;
   i = strcmp ( string1, "Jerry" ) ;
   j = strcmp ( string1, string2 ) ;
   k = strcmp ( string1, "Jerry boy" ) ;
   printf ( "\n%d %d %d", i, j, k ) ;
}
```

And here is the output...
0 4 -32

strcmp( )

- In the first call to **strcmp( )**, the two strings are identical—“Jerry” and “Jerry”—and the value returned by **strcmp( )** is zero.
- In the second call, the first character of “Jerry” doesn't match with the first character of “Ferry” and the result is 4, which is the numeric difference between ASCII value of ‘J’ and ASCII value of ‘F’.
- In the third call to **strcmp( )** “Jerry” doesn’t match with “Jerry boy”, because the null character at the end of “Jerry” doesn’t match the blank in “Jerry boy”.
- The value returned is -32, which is the value of null character minus the ASCII value of space, i.e., ‘\0’ minus ‘ ’, which is equal to -32.
- Any non-zero value means there is a mismatch.

Summary

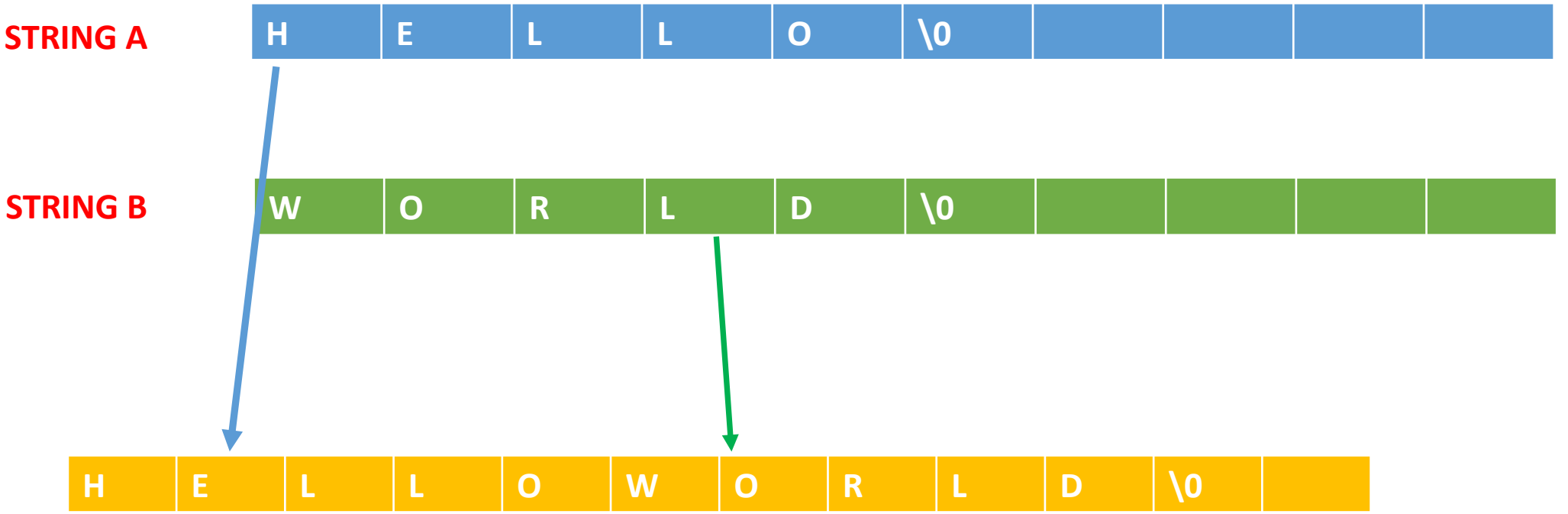
- A string is nothing but an array of characters terminated by ‘\0’.
- Being an array, all the characters of a string are stored in contiguous memory locations.
- Though **scanf( )** can be used to receive multi-word strings, **gets( )** can do the same job in a cleaner way.
- Both **printf( )** and **puts( )** can handle multi-word strings.
- Strings can be operated upon using several standard library functions like **strlen( )**, **strcpy( )**, **strcat( )** and **strcmp( )** which can manipulate strings.

Exercise

- Write a program that converts all lowercase characters in a given string to its equivalent uppercase character.
- [Hint: There is an inbuilt function to change the case.]

- Write a program to concatenate two strings (WITHOUT USING INBUILT FUNCTION)
- Write a user defined program to compare two strings
- Write a user defined program to reverse a string without using a separate array
- Write a user defined program to remove all occurrences of vowels

Write a program to concatenate two strings



Write a program to concatenate two strings

- Input the First String1
- Input the Second String2
- Find the length of first string
- Copy the string 2 from the position after the last element of string 1
- Put Null character at the end of String

Write a program to concatenate two strings


```
#include <string.h>
int main()
{
    char s1[1000],s2[1000];
    int i,j;
    printf("Enter  string1: ");
     gets(s1);
    printf("Enter  string2: ");
    gets(s2);
    j=strlen(s1);

    for(i=0;s2[i]!='\0';i++)
    {
     s1[i+j]=s2[i];
}
s1[i+j]='\0';
    printf("combined two strings ='%s'\n",s1);



    return 0;
}
```

Input the First String1

Input the Second String2

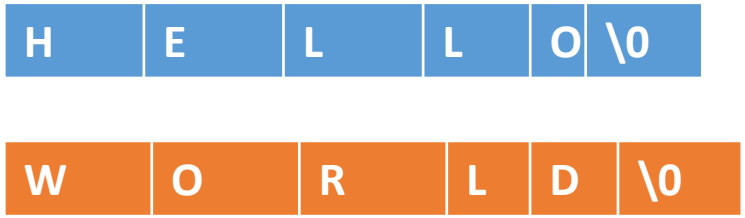
Find the length of first string

Copy the string 2 from the position after the last element of string 1

Put Null character at the end of String

Write a program to concatenate two strings

```
#include <string.h>
int main()
{
    char s1[1000],s2[1000];
    int i,j;
    printf("Enter  string1: ");
    gets(s1);
    printf("Enter  string2: ");
     gets(s2);
    j=strlen(s1);

    for(i=0;s2[i]!='\0';i++)
    {
     s1[i+j]=s2[i];
}
s1[i+j]='\0';
    printf("combined two strings ='%s'\n",s1);



    return 0;
}
```



Input the First String1

Input the Second String2

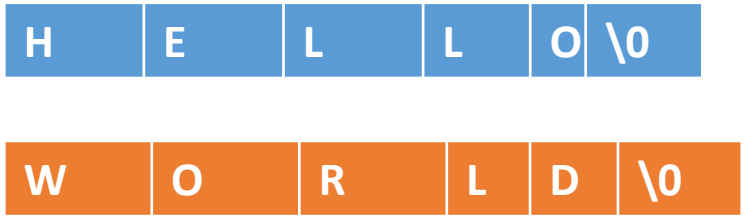
Find the length of first string

Copy the string 2 from the position after the last element of string 1

Put Null character at the end of String

Write a program to concatenate two strings

```
#include <string.h>
int main()
{
    char s1[1000],s2[1000];
    int i,j;
    printf("Enter  string1: ");
    gets(s1);
    printf("Enter  string2: ");
    gets(s2);
    j=strlen(s1);

    for(i=0;s2[i]!='\0';i++)
    {
     s1[i+j]=s2[i];
}
s1[i+j]='\0';
    printf("combined two strings ='%s'\n",s1);



    return 0;
}
```



Input the First String1

J= LENGTH OF STRING1 = 5

Input the Second String2

Find the length of first string

Copy the string 2 from the position after the last element of string 1

Put Null character at the end of String

Write a program to concatenate two strings

```
#include <string.h>
int main()
{
    char s1[1000],s2[1000];
    int i,j;
    printf("Enter  string1: ");
    gets(s1);
    printf("Enter  string2: ");
    gets(s2);
    j=strlen(s1);

    for(i=0;s2[i]!='\0';i++)
     {
      s1[i+j]=s2[i];
}
s1[i+j]='\0';
    printf("combined two strings ='%s'\n",s1);



    return 0;
}
```

J= LENGTH OF STRING1 = 5

Input the First String1

Input the Second String2

Find the length of first string

Copy the string 2 from the position after the last element of string 1

Put Null character at the end of String

I=0, J=5	S1[I+J]
I=0	S1[5]=W

Write a program to concatenate two strings

J= LENGTH OF STRING1 = 5


```
#include <string.h>
int main()
{
    char s1[1000],s2[1000];
    int i,j;
    printf("Enter  string1: ");
    gets(s1);
    printf("Enter  string2: ");
    gets(s2);
    j=strlen(s1);

    for(i=0;s2[i]!='\0';i++)
    {
     s1[i+j]=s2[i];
}
s1[i+j]='\0';
    printf("combined two strings ='%s'\n",s1);



    return 0;
}
```

Input the First String1

Input the Second String2

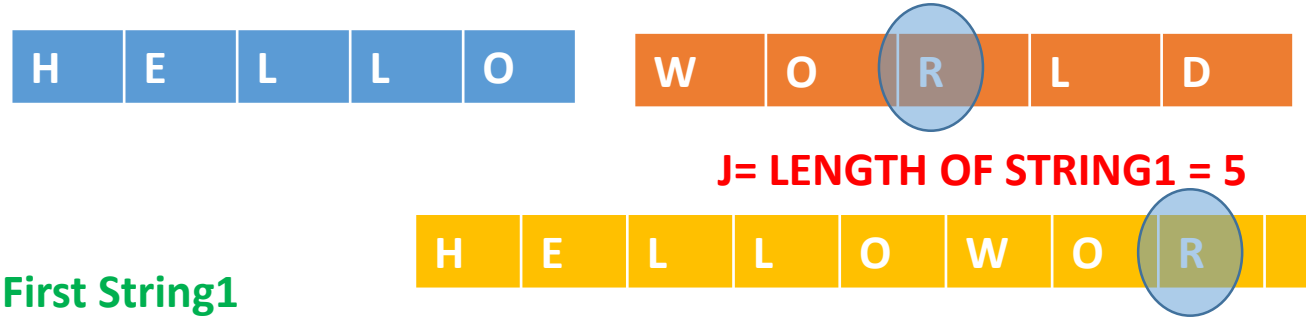
Find the length of first string

Copy the string 2 from the position after the last element of string 1

Put Null character at the end of String

I=0, J=5	S1[I+J]
I=0	S1[5]=W
I=1	S1[6]=O

Write a program to concatenate two strings




```c
#include <string.h>
int main()
{
    char s1[1000],s2[1000];
    int i,j;
    printf("Enter  string1: ");
    gets(s1);
    printf("Enter  string2: ");
    gets(s2);
    j=strlen(s1);

    for(i=0;s2[i]!='\0';i++)
    {
     s1[i+j]=s2[i];
}
s1[i+j]='\0';
    printf("combined two strings ='%s'\n",s1);


    return 0;
}
```

Input the First String1

Input the Second String2

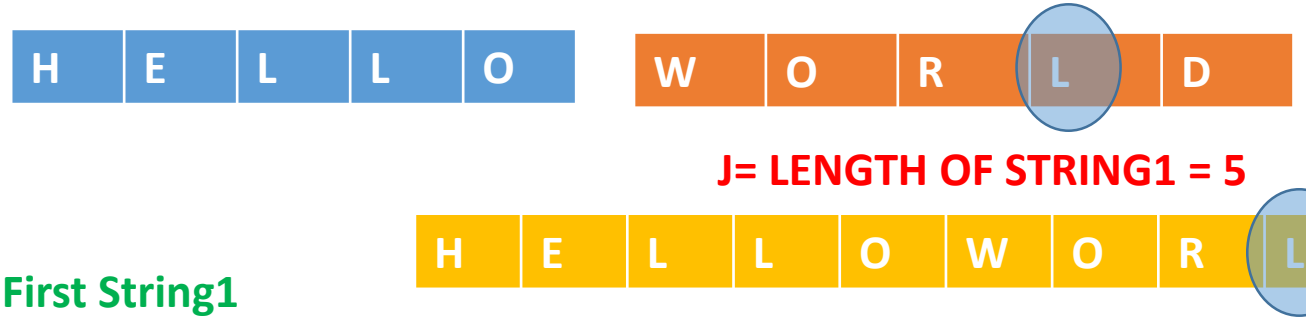
Find the length of first string

Copy the string 2 from the position after the last element of string 1

Put Null character at the end of String

I=0, J=5	S1[I+J]
I=0	S1[5]=W
I=1	S1[6]=O
I=2	S1[7]=R

Write a program to concatenate two strings




```c
#include <string.h>
int main()
{
    char s1[1000],s2[1000];
    int i,j;
    printf("Enter  string1: ");
    gets(s1);
    printf("Enter  string2: ");
    gets(s2);
    j=strlen(s1);

    for(i=0;s2[i]!='\0';i++)
    {
     s1[i+j]=s2[i];
}
s1[i+j]='\0';
    printf("combined two strings ='%s'\n",s1);


    return 0;
}
```

Input the First String1

Input the Second String2

Find the length of first string

Copy the string 2 from the position after the last element of string 1

Put Null character at the end of String

I=0, J=5	S1[I+J]
I=0	S1[5]=W
I=1	S1[6]=O
I=2	S1[7]=R
I=3	S1[8]=L

Write a program to concatenate two strings

```
#include <string.h>
int main()
{
    char s1[1000],s2[1000];
    int i,j;
    printf("Enter  string1: ");
    gets(s1);
    printf("Enter  string2: ");
    gets(s2);
    j=strlen(s1);

    for(i=0;s2[i]!='\0';i++)
    {
     s1[i+j]=s2[i];
}
s1[i+j]='\0';
    printf("combined two strings ='%s'\n",s1);


    return 0;
}
```

J= LENGTH OF STRING1 = 5

Input the First String1

Input the Second String2

Find the length of first string

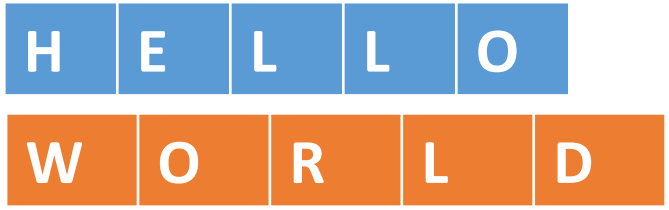
Copy the string 2 from the position after the last element of string 1

Put Null character at the end of String

I=0, J=5	S1[I+J]
I=0	S1[5]=W
I=1	S1[6]=O
I=2	S1[7]=R
I=3	S1[8]=L
I=4	S1[9]=D

Write a program to concatenate two strings

```
#include <string.h>
int main()
{
    char s1[1000],s2[1000];
    int i,j;
    printf("Enter  string1: ");
    gets(s1);
    printf("Enter  string2: ");
    gets(s2);
    j=strlen(s1);

    for(i=0;s2[i]!='\0';i++)
    {
     s1[i+j]=s2[i];
}
s1[i+j]='\0';
    printf("combined two strings ='%s'\n",s1);


    return 0;
}
```



Input the First String1

Input the Second String2

Find the length of first string

LENGTH OF STRING1 = 5

Copy the string 2 from the position after the last element of string 1

Put Null character at the end of String

Write a program to compare two strings using Inbuilt Function

- Input two strings
- Use Inbuilt strcmp function

Write a program to compare two strings using inbuilt function

```c
#include <stdio.h>
#include <string.h>
int main()
{
    char a[100], b[100];
    printf("Enter a string\n");
    gets(a);
    printf("Enter a string\n");
    gets(b);
    if (strcmp(a,b) == 0)
       printf("The strings are equal.\n");
    else
       printf("The strings are not equal.\n");
    return 0;
}
```

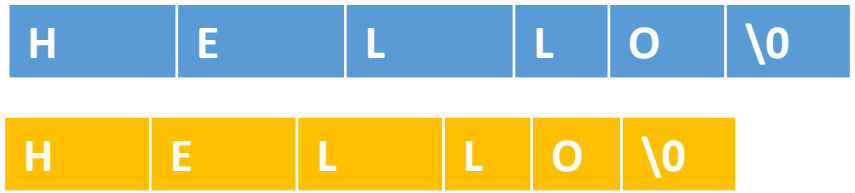
Write a program to compare two strings using User defined Function

- Input two strings
- Keep comparing as long as the two strings match
  - Terminate when either of the two strings is equal to Null
  - OR there is a mismatch
- Check whether both the strings have null character
  - If Both the strings match, print 0
- else
  - strings are Dissimilar and print the difference in ascii value of string 1 and string 2

Write a program to compare two strings using user defined function

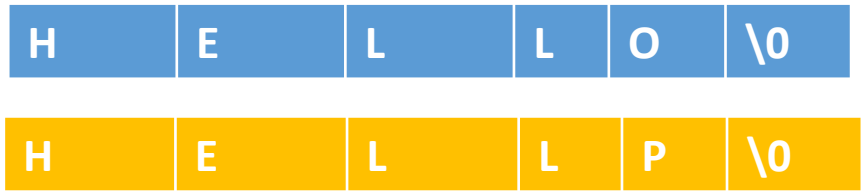
```
#include<stdio.h>
#include<string.h>
int main()
{
        char string1[100], string2[100],c=0,t;
        printf("Enter first string");
        gets(string1);
        printf("Enter Second string");
        gets(string2);
        while(string1[c]==string2[c])
        {
                if(string1[c]=='\0'&& string2[c]=='\0')
                 break;
                else
                c++;
        }
```

```
if(string1[c]=='\0'&&string2[c]=='\0')
        t=0;
        else
        t=string1[c] – string2[c];

        printf("%d",t);
                return 0;
}
```



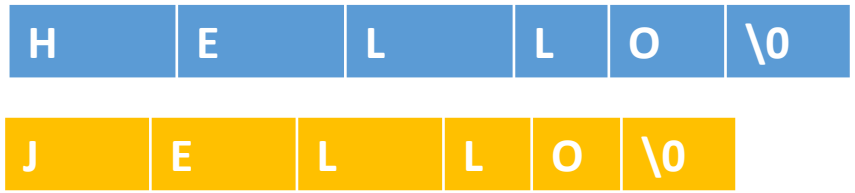
Write a program to compare two strings using user defined function

```
#include<stdio.h>
#include<string.h>
int main()
{
        char string1[100], string2[100],c=0,t;
        printf("Enter first string");
        gets(string1);
        printf("Enter Second string");
        gets(string2);
        while(string1[c]==string2[c])
        {
                if(string1[c]=='\0'&& string2[c]=='\0')
                 break;
                else
                c++;
        }
        if(string1[c]=='\0'&&string2[c]=='\0')
                t=0;
                else
                t=string1[c] – string2[c];

                printf("%d",t);
                        return 0;
}
```



Write a program to compare two strings using user defined function

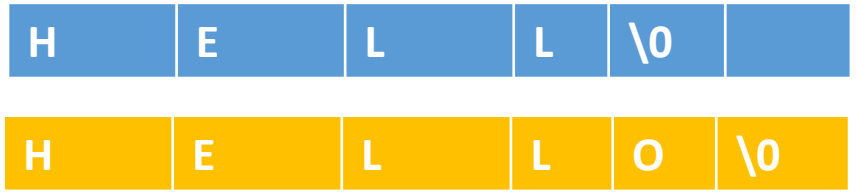
```
#include<stdio.h>
#include<string.h>
int main()
{
        char string1[100], string2[100],c=0,t;
        printf("Enter first string");
        gets(string1);
        printf("Enter Second string");
        gets(string2);
        while(string1[c]==string2[c])
        {
                if(string1[c]=='\0'&& string2[c]=='\0')
                 break;
                else
                c++;
        }
        if(string1[c]=='\0'&&string2[c]=='\0')
                        t=0;
                        else
                          t=string1[c] – string2[c];

                        printf("%d",t);
                                return 0;
}
```



Write a program to compare two strings using user defined function

```
#include<stdio.h>
#include<string.h>
int main()
{
        char string1[100], string2[100],c=0,t;
        printf("Enter first string");
        gets(string1);
        printf("Enter Second string");
        gets(string2);
        while(string1[c]==string2[c])
        {
                if(string1[c]=='\0'&& string2[c]=='\0')
                 break;
                else
                c++;
        }
```

```
if(string1[c]=='\0'&&string2[c]=='\0')
                t=0;
                else
                t=string1[c] – string2[c];

                printf("%d",t);
                        return 0;
}
```



Write a program to reverse a string without using a separate array

- Input the string
- Using Inbuilt function find the length of the string
- Traverse the array till half of its length
- Initialize End =n-1 (which is last element of Array)
- Using loop perform the step
  - Temp=Arr[i]
  - Arr[i] = Arr[end]
  - Arr[end]=Temp
  - i=i+1, End =End-1

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
L	I	F	E	I	S	B	E	A	U	T	I	F	U	L

t =15

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
L	I	F	E	I	S	B	E	A	U	T	I	F	U	L

n=14

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
L	I	F	E	I	S	B	E	A	U	T	I	F	U	L

n=14

t= 6

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{
        char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

Swap 0th element with n-1th element
Swap 1st element with n-2th element
Swap 2nd element with n-3rd element
.
.
.

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
L	I	F	E	I	S	B	E	A	U	T	I	F	U	L

n=14

t= 6

C=0	N=14

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
L	I	F	E	I	S	B	E	A	U	T	I	F	U	L

n=14

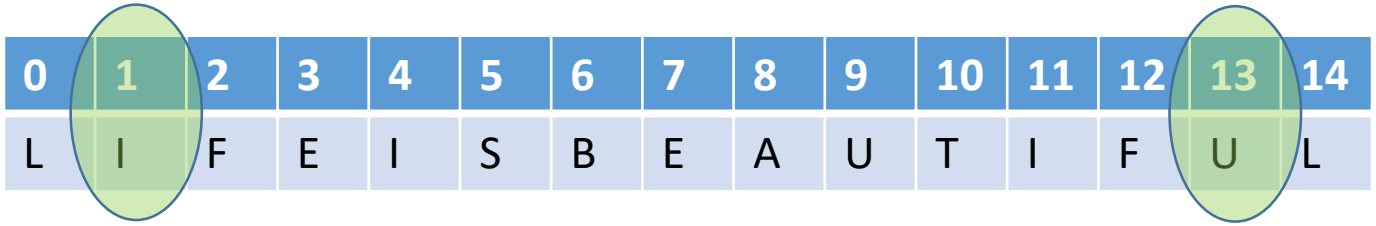
t= 6

C=0	N=14

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
L	I	F	E	I	S	B	E	A	U	T	I	F	U	L

n=14

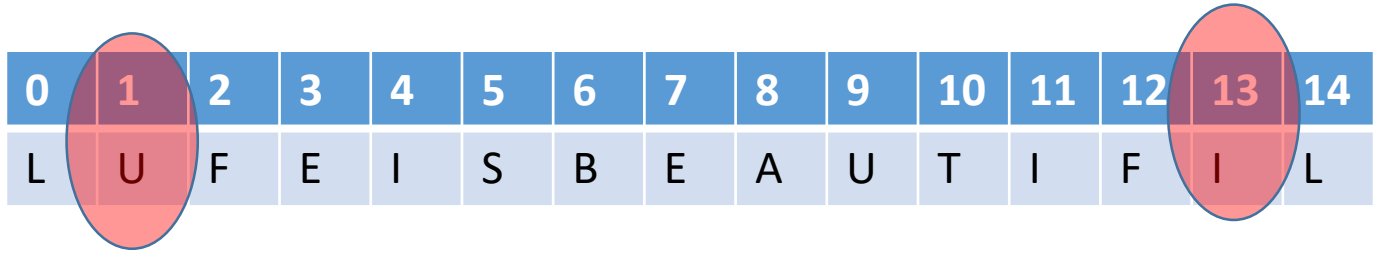
t= 6

C=0	N=14
C=1	N=13

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],temp;
Int c,n,t;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed



n=14

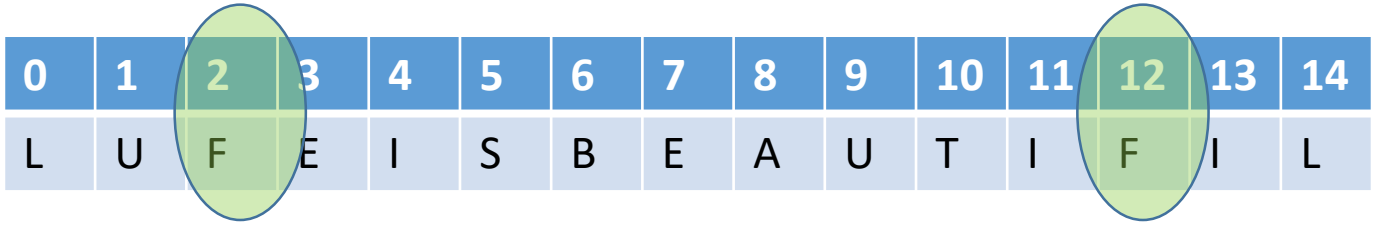
t= 6

C=0	N=14
C=1	N=13

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
L	U	F	E	I	S	B	E	A	U	T	I	F	I	L

n=14

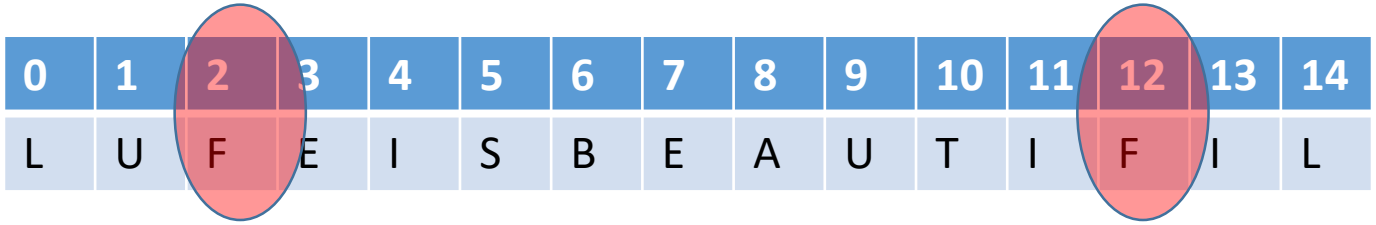
t= 6

C=0	N=14
C=1	N=13
C=2	N=12

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
L	U	F	E	I	S	B	E	A	U	T	I	F	I	L

n=14

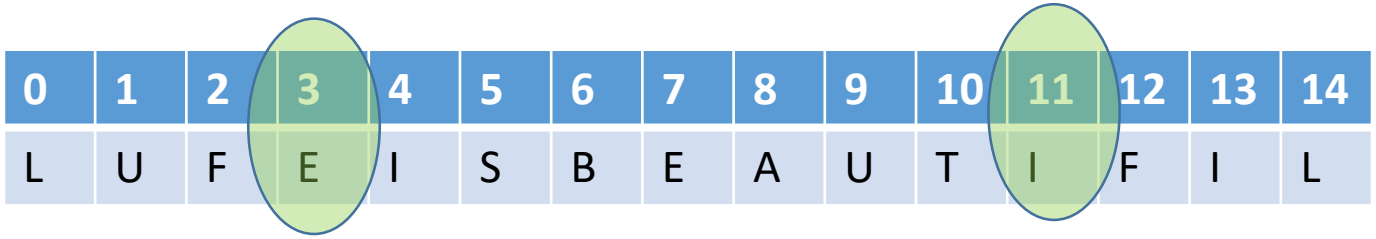
t= 6

C=0	N=14
C=1	N=13
C=2	N=12

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed



n=14

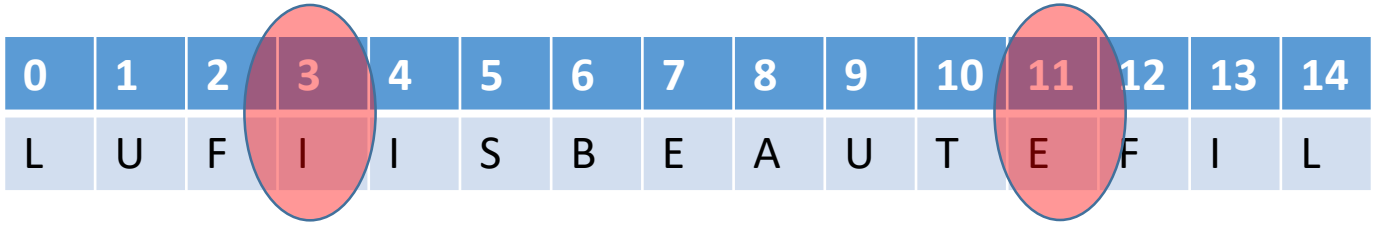
t= 6

C=0	N=14
C=1	N=13
C=2	N=12
C=3	N=11

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed



n=14

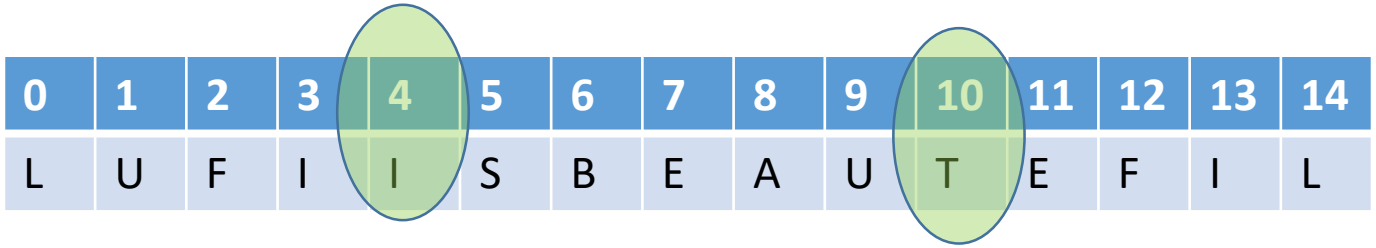
t= 6

C=0	N=14
C=1	N=13
C=2	N=12
C=3	N=11

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed



n=14

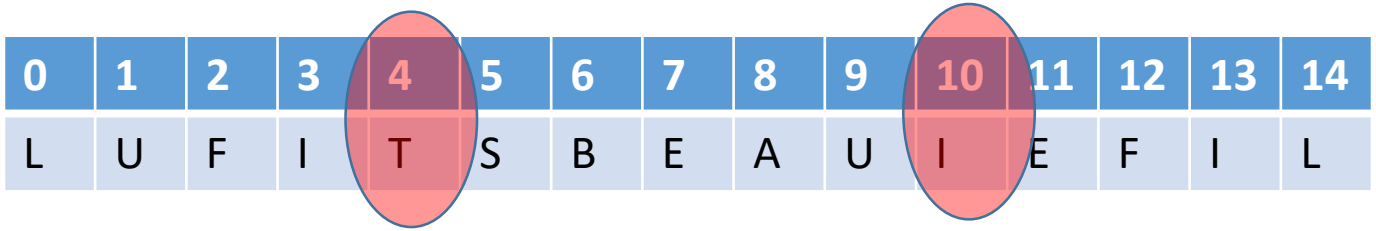
t= 6

C=0	N=14
C=1	N=13
C=2	N=12
C=3	N=11
C=4	N=10

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed



n=14

t= 6

C=0	N=14
C=1	N=13
C=2	N=12
C=3	N=11
C=4	N=10

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
L	U	F	I	T	S	B	E	A	U	I	E	F	I	L

n=14

t= 6

C=0	N=14
C=1	N=13
C=2	N=12
C=3	N=11
C=4	N=10
C=5	N=9

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
L	U	F	I	T	U	B	E	A	S	I	E	F	I	L

n=14

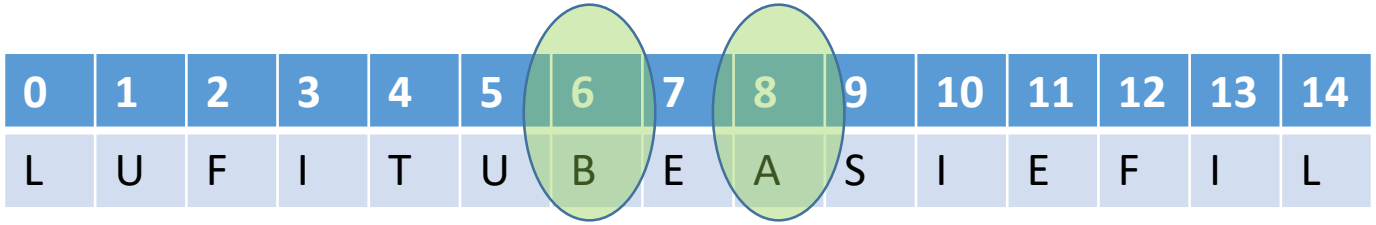
t= 6

C=0	N=14
C=1	N=13
C=2	N=12
C=3	N=11
C=4	N=10
C=5	N=9

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
L	U	F	I	T	U	B	E	A	S	I	E	F	I	L

n=14

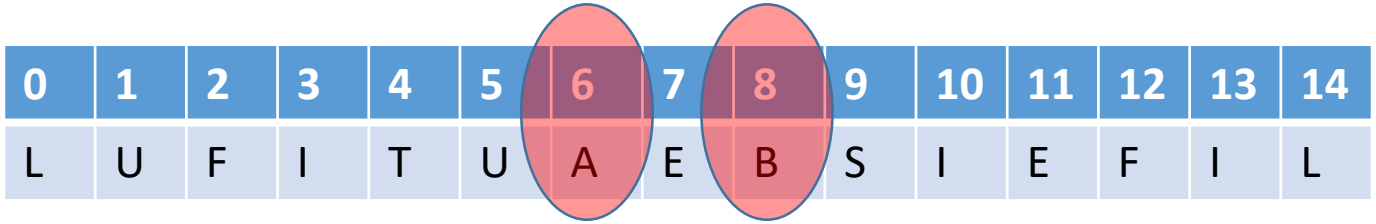
t= 6

C=0	N=14
C=1	N=13
C=2	N=12
C=3	N=11
C=4	N=10
C=5	N=9
C=6	N=8

Write a program to reverse a string without using a separate array

```
#include<stdio.h>
#include<string.h>
int main()
{       char a[100],c=0,n,t,temp;
        printf("Enter first string");
        gets(a);
        t=strlen(a);
        n=t-1;
        t=t/2-1;
        for(c=0;c<=t;c++)
        {
                temp=a[c];
                a[c]= a[n];
                a[n]=temp;
                n--;
        }
        puts(a);
}
```

After finding the length , Set t equal to half its length, so that half of array is traversed



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
L	U	F	I	T	U	A	E	B	S	I	E	F	I	L

n=14

t= 6

C=0	N=14
C=1	N=13
C=2	N=12
C=3	N=11
C=4	N=10
C=5	N=9
C=6	N=8

Write a program to remove all occurrences of vowels

I=0

J=0

0	1	2	3	4	5	6	7	8	9	10	11
M	I	S	S	I	S	I	P	P	I	A	N

I=1

J=1

0	1	2	3	4	5	6	7	8	9	10	11
M											

I=1

J=1

0	1	2	3	4	5	6	7	8	9	10	11
M	I	S	S	I	S	I	P	P	I	A	N

I=2

J=1

0	1	2	3	4	5	6	7	8	9	10	11
M											

I=2

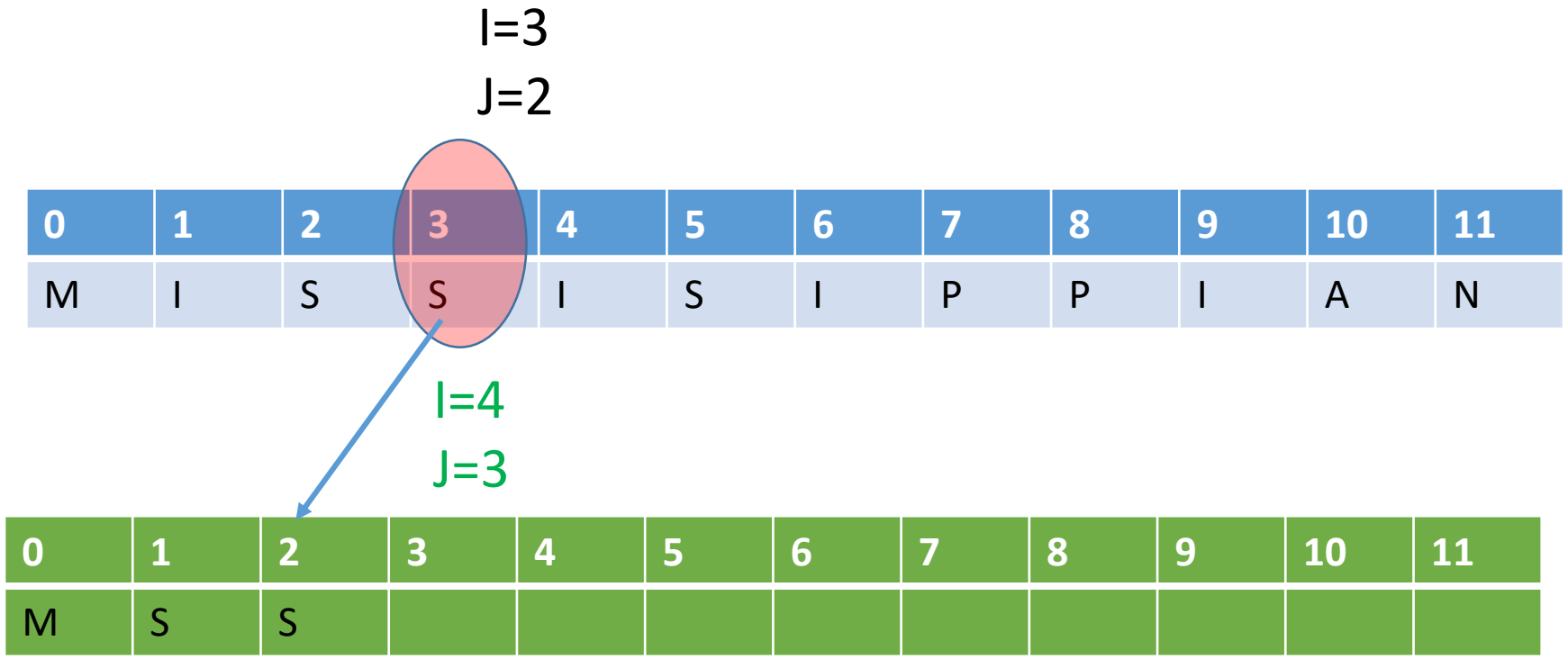
J=1

0	1	2	3	4	5	6	7	8	9	10	11
M	I	S	S	I	S	I	P	P	I	A	N

I=3

J=2

0	1	2	3	4	5	6	7	8	9	10	11
M	S										


I=3
J=2
0 1 2 3 4 5 6 7 8 9 10 11
M I S S I S I P P I A N
I=4
J=3
0 1 2 3 4 5 6 7 8 9 10 11
M S S

I=4

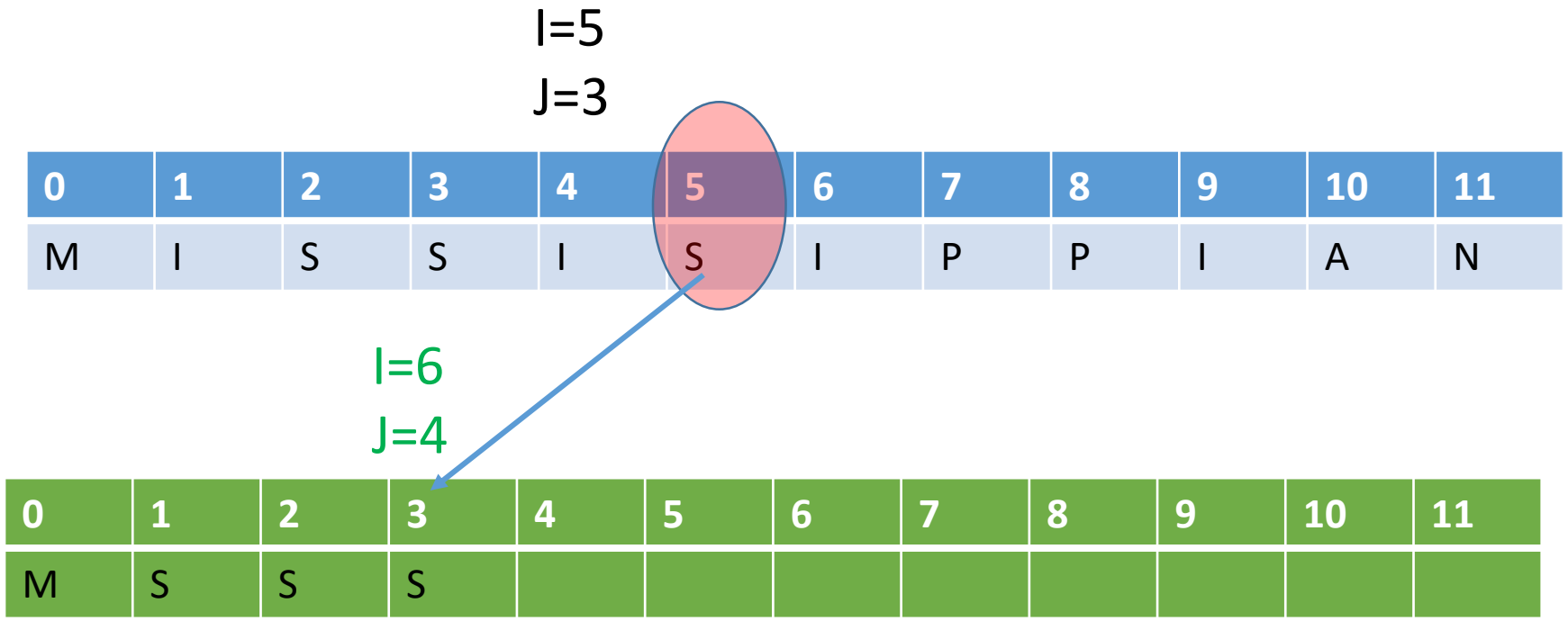
J=3

0	1	2	3	4	5	6	7	8	9	10	11
M	I	S	S	I	S	I	P	P	I	A	N

I=5

J=3

0	1	2	3	4	5	6	7	8	9	10	11
M	S	S									


I=5
J=3
0 1 2 3 4 5 6 7 8 9 10 11
M I S S I S I P P I A N
I=6
J=4
0 1 2 3 4 5 6 7 8 9 10 11
M S S S

I=6

J=4

0	1	2	3	4	5	6	7	8	9	10	11
M	I	S	S	I	S	I	P	P	I	A	N

I=7

J=4

0	1	2	3	4	5	6	7	8	9	10	11
M	S	S	S								

I=7

J=4

0	1	2	3	4	5	6	7	8	9	10	11
M	I	S	S	I	S	I	P	P	I	A	N

I=8

J=5

0	1	2	3	4	5	6	7	8	9	10	11
M	S	S	S	P							

I=8

J=5

0	1	2	3	4	5	6	7	8	9	10	11
M	I	S	S	I	S	I	P	P	I	A	N

I=9

J=6

0	1	2	3	4	5	6	7	8	9	10	11
M	S	S	S	P	P						

I=9

J=6

0	1	2	3	4	5	6	7	8	9	10	11
M	I	S	S	I	S	I	P	P	I	A	N

I=10

J=6

0	1	2	3	4	5	6	7	8	9	10	11
M	S	S	S	P	P						

I=10

J=6

0	1	2	3	4	5	6	7	8	9	10	11
M	I	S	S	I	S	I	P	P	I	A	N

I=11

J=6

0	1	2	3	4	5	6	7	8	9	10	11
M	S	S	S	P	P						

I=11
J=6

0	1	2	3	4	5	6	7	8	9	10	11
M	I	S	S	I	S	I	P	P	I	A	N

I=11
J=7

0	1	2	3	4	5	6	7	8	9	10	11
M	S	S	S	P	P	N					

I=11

J=6

0	1	2	3	4	5	6	7	8	9	10	11
M	I	S	S	I	S	I	P	P	I	A	N

I=11

J=7

0	1	2	3	4	5	6	7	8	9	10	11
M	S	S	S	P	P	N	\0				

```
J=0;
for(i=0;a[i]!=NULL;i++)
 {
    if(a[i] is not equal to any of vowels)
    {  a[j]=a[i]
       j++
 }  }
```