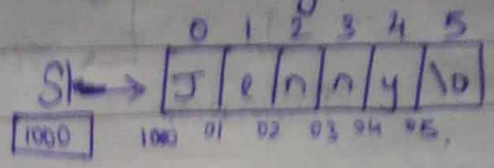
C_67 => Program to Concatenate two strings

* Concatenation means joining two strings.

* For this in which source string we are going to join the other (or) destination string; then the size of source string must be larger size.

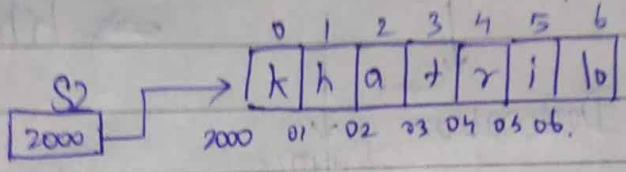
Eg:

Char S1[6] = "Jenny";



6 bytes

Char S2[7] = "Khatri";



7 bytes

* To concatinate (or) add S2[] in S1[], then size of S1[] should be larger.

* So instead of S1[6]; we take S1[30].

```
Char S1[30] = "Jenny";
Char S2[7] = "khatri";
```

* Predefined function to concatenate 2 strings, which is already defined in string.h

strcat (Argument 1, Argument 2)
Destination Source.

Argument 2 → Source.

Argument 1 → Destination

* This function will return pointer to the destination string

* Always first argument will be destination string.

* strcat (S1, S2); ⇒ Jenny Khatri
  - S1 → Destination
  - S2 → source.
  - no space

* strcat (S2, S1); ⇒ Khatri Jenny.
  - S2 → Destination
  - S1 → Source.
  - no space.

* **Buffer overflow problem :-**

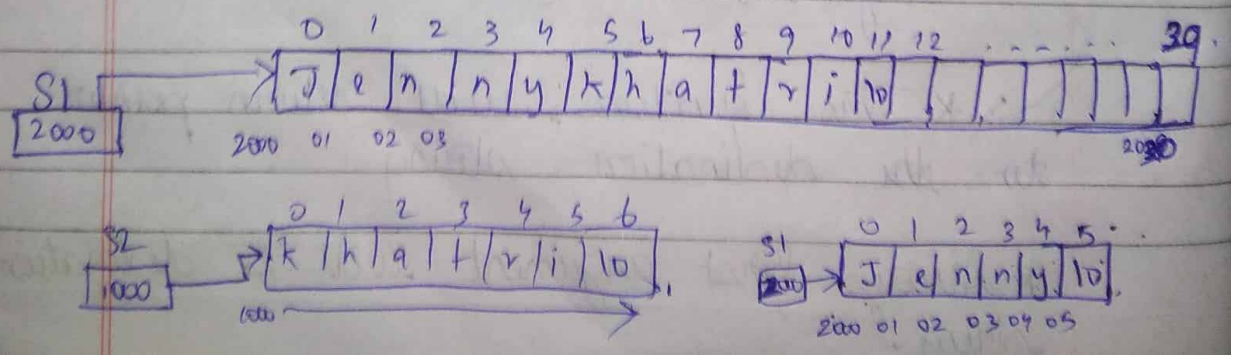
* Eventhough the array size of the string destination is small; this Strcat() will concate the source string without checking the buffer size.

Program :- [using strcat() function]

```
#include <string.h>
void main()
{
    char S1[30] = "Jenny";
    char S2[7] = "Khatri";
    strcat(S1, S2);
    printf("%.S", S1);
}
```

o/p

Jenny Khatri



Program 2 [without using strcat() function]

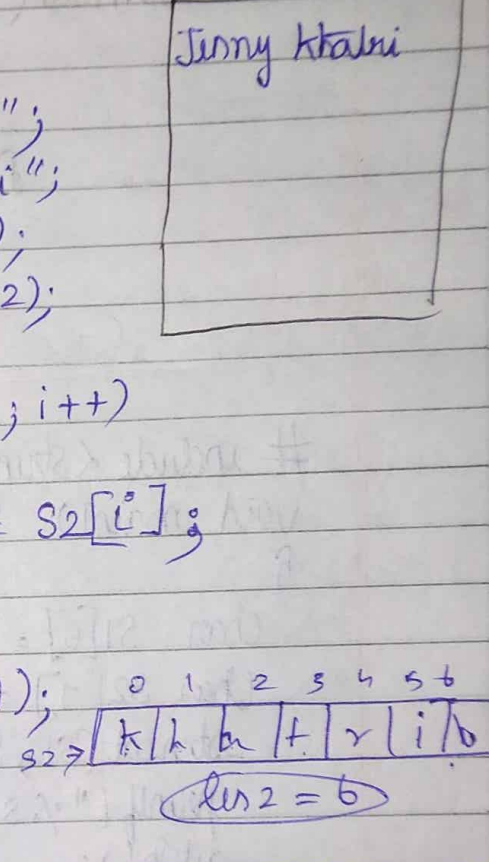
* For this we should know the length of a string source & destination.

```
void main()
{
    int len1, len2;
    char S1[30] = "Jenny ";
    char S2[7] = "Khatri";
      len1 = strlen(S1);
      len2 = strlen(S2);

    for(i=0; i<=len2; i++)
    {
      S1[len1+i] = S2[i];
    }
    printf("%s", S1);
}
```

O/P

Jenny Khatri



from this S1[5] ⇒ S1[len1+i] (starting) we store S2[i] values.

Note: Check; if we want to Concatenate only 4 characters of S2; then i<4 and print the S1 value ; output; because we are not including null character at end.

Program 3

* When we want to Concatenate only n characters of source into destination by using string function, we use.

strncat(Argument 1, Argument 2, Argument 3);

Argument 1 → Destination
Argument 2 → Source
Argument 3 → No. of characters from source.

```
# include <string.h>
void main()
{
  char S1[6] = "Jenny";
  char S2[7] = "khatri";
  strncat(S1, S2, 4);
  printf("%s", S1);
  getch();
}
```

O/P

Jennykha.

Note:

Strncat( )

↳ * n → Concatenate upto n characters.

Note:

* String will automatically put null character at end of string in memory.

* But when we use without string concatenate function we should also include the null character when with concatenate string; to get the output with end of string.

* strcat() function default will have this null character.

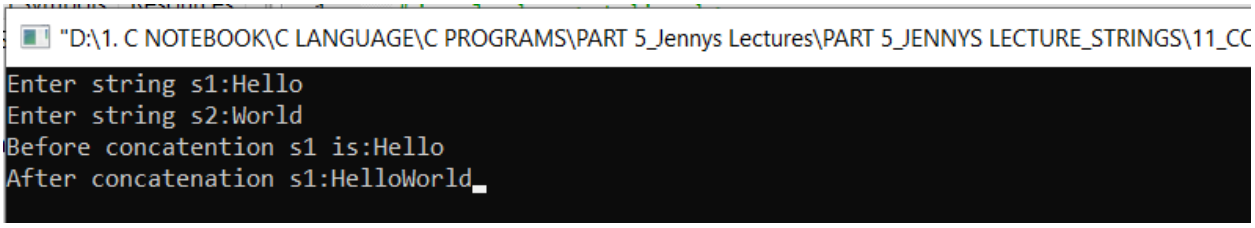
* strcat() will have buffer overflow problem (i.e) eventhough the destination size is small, it will add the strings of source beyond the size of distination.

* In strcat(), always the distination string should be written first followed by source string.

* strncat() will have destination, Sour and n number of characters to be included in destination string.

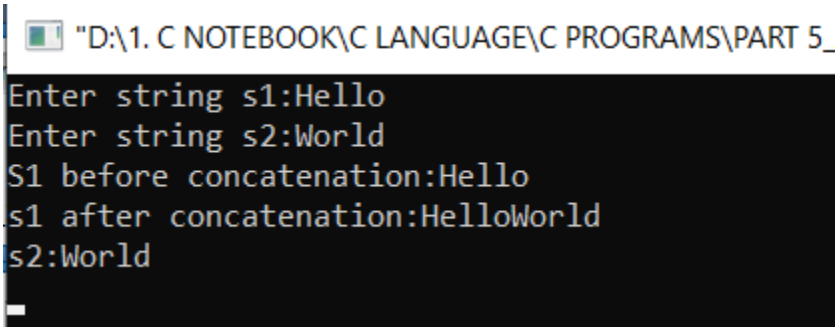
CODE 1:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
/** 1 - STRING CONCATENATE **/
/** ADDING SOURCE STRING TO DESTINATION STRING **/
/** USING strcat() function **/
int main()
{
    char s1[30],s2[6];
    printf("Enter string s1:");
    gets(s1);
    printf("Enter string s2:");
    gets(s2);
    printf("Before concatention s1 is:");
    puts(s1);
    strcat(s1,s2);
    printf("After concatenation s1:%s",s1);
    getch();
}
/** Destination string should have the larger size for concatenation **/
/** Here s1 is destination string and it should have large size **/
```



CODE 2:

```c
#include <stdio.h>
#include <stdlib.h>
/** 2 - STRING CONCATENATE **/
/** ADDING SOURCE STRING TO DESTINATION STRING **/
/** without using strcat() function, using for loop **/
int main()
{
    char s1[30],s2[6];
    int i,length_1,length_2;
    printf("Enter string s1:");
    gets(s1);
    printf("Enter string s2:");
    gets(s2);
    printf("S1 before concatenation:");
    puts(s1);
    length_1=strlen(s1);
    length_2=strlen(s2);
    for(i=0;i<=length_2;i++) //null character included
    {
     s1[length_1+i]=s2[i];
    }
    printf("s1 after concatenation:");
    puts(s1);
    printf("s2:");
    puts(s2);
    getch();
}
```



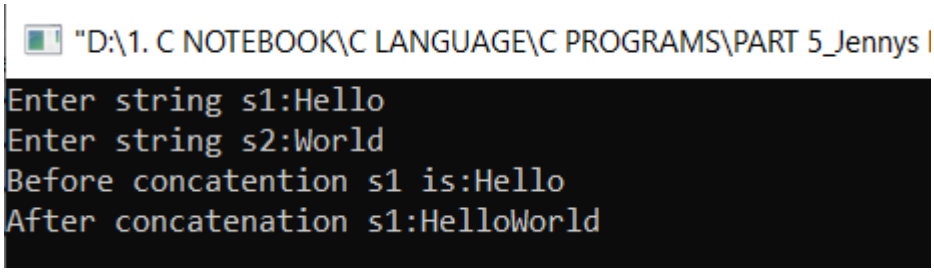
```c
#include <stdio.h>
#include <stdlib.h>
/** 2 - STRING CONCATENATE **/
/** ADDING SOURCE STRING TO DESTINATION STRING **/
/** without using strcat() function, using for loop **/
int main()
{
    char s1[30],s2[6];
    int i,length_1,length_2;
    printf("Enter string s1:");
    gets(s1);
    printf("Enter string s2:");
    gets(s2);
    printf("S1 before concatenation:");
    puts(s1);
    length_1=strlen(s1);
    length_2=strlen(s2);
    for(i=0;i<length_2;i++) //null character not included
    {
     s1[length_1+i]=s2[i];
    }
    printf("s1 after concatenation:");
    puts(s1);
    printf("s2:");
    puts(s2);
    getch();
}
```

```
"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures
Enter string s1:Hello
Enter string s2:World
S1 before concatenation:Hello
s1 after concatenation:HelloWorld@
s2:World
```

CODE 3:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
/** 3 - STRING CONCATENATE **/
/** ADDING SOURCE STRING TO DESTINATION STRING **/
/** USING strcat() function **/
/** Buffer overflow with strcat() function **/
int main()
{
    char s1[6],s2[6];
    printf("Enter string s1:");
    gets(s1);
    printf("Enter string s2:");
    gets(s2);
    printf("Before concatention s1 is:");
    puts(s1);
    strcat(s1,s2);
    printf("After concatenation s1:%s",s1);
    getch();
}
```

```c
/** Destination string should have the larger size for concatenation **/
/** Here s1 is destination string is only 6 but needs more size for
    concatenation, even though size is small, strcat() will add string
    beyond the limit which leads to buffer overflow problem..  **/
```



CODE 4:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
/** 4 - STRING CONCATENATE **/
/** ADDING upto 'n' number characters of SOURCE STRING TO DESTINATION STRING **/
/** USING strncat() function with 3 arguments **/
/** strcat(destination,source,n) **/
int main()
{
    char s1[30],s2[6];
    printf("Enter string s1:");
    gets(s1);
    printf("Enter string s2:");
    gets(s2);
    printf("Before concatention s1 is:");
    puts(s1);
    strncat(s1,s2,3);
    printf("After concatenation s1:%s",s1);
    getch();
}
/** From source string only upto 3 characters are added to destination string**/
```