**String Append**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define SIZE 20

char \*strappend(char\*, char\*);

int main()

{

char \*src = NULL;

char \*dest = NULL;

src = (char\*)malloc(SIZE - sizeof(char));

dest = (char\*)malloc(SIZE - sizeof(char));

if (NULL == src && NULL == dest) {

printf("Malloc failed!!\n");

exit(0);

}

printf("Enter the first string: ");

if(NULL == (fgets(src, SIZE, stdin))) {

printf("ERROR!!!");

}

printf("Enter the second string: ");

if(NULL == (fgets(dest, SIZE, stdin))) {

printf("ERROR!!!");

}

\*(src + (strlen(src)) - 1) = '\0';

\*(dest + (strlen(dest)) - 1) = '\0';

strappend(src, dest);

printf("string is :%s\n",src);

free(src);

free(dest);

src = NULL;

dest = NULL;

}

char \*strappend(char \*src,char \*dest)

{

while(\*src != '\0') {

src++;

}

while(\*dest != '\0') {

\*src = \*dest ;

\*src++;

\*dest++;

}

\*src == '\0';

return src;

}

**String Cat**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define SIZE 20

char\* mystrcat(char \*src1, char \*src2);

int main()

{

char \*src1;

char \*src2;

src1 = (char \*) malloc (SIZE \* sizeof(char));

src2 = (char \*) malloc (SIZE \* sizeof(char));

if (NULL == src1 && NULL == src2) {

printf("malloc failed!!\n");

exit(0);

}

printf("Enter the first String: \n");

if (NULL == (fgets(src1, SIZE, stdin))) {

printf("Fgets failed for src1");

}

printf("Enter the second String: \n");

if (NULL == (fgets(src2, SIZE, stdin))) {

printf("Fgets failed for src2");

}

\*(src1 + (strlen(src1)) - 1) = '\0';

\*(src2 + (strlen(src2)) - 1) = '\0';

printf("Concatinated String is %s\n", mystrcat(src1, src2));

free(src1);

free(src2);

src1 = NULL;

src2 = NULL;

}

char\* mystrcat(char \*src1, char \*src2)

{

char \*temp = src1;

while (\*src1) {

src1++;

}

while (\*src2) {

\*src1++ = \*src2++;

}

\*src1 = '\0';

return temp;

}

* **String Compare**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define SIZE 20

int strcomp(char\*, char\*);

int main()

{

char \*str1 = NULL;

char \*str2 = NULL;

str1 = (char\*) malloc (SIZE - sizeof(char));

str2 = (char\*) malloc (SIZE - sizeof(char));

if (NULL == str1 && NULL == str2) {

printf("Malloc failed!!\n");

exit(0);

}

printf("Enter the first string: ");

if(NULL == (fgets(str1, SIZE, stdin))) {

printf("ERROR!!!");

}

printf("Enter the second string: ");

if(NULL == (fgets(str2, SIZE, stdin))) {

printf("ERROR!!!");

}

\*(str1 + (strlen(str1)) - 1) = '\0';

\*(str2 + (strlen(str2)) - 1) = '\0';

printf("%d\n", strcomp(str1, str2));

free(str1);

free(str2);

str1 = NULL;

str2 = NULL;

return 0;

}

int strcomp(char \*str1,char \*str2)

{

while(\*str1 != '\0' && \*str2!= '\0') {

if(\*str1 == \*str2 && (strlen(str1) == strlen(str2))) {

// \*str1++;

//\*str2++;

return 0;

} else if(\*str1 == \*str2 && (strlen(str1) > strlen(str2))) {

return 1;

} else if(\*str1 == \*str2 && (strlen(str1) < strlen(str2))) {

return -1;

} else {

printf("Strings are not equal\n");

return 2;

}

\*str1++;

\*str2++;

}

}

* **String CaseCompare**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <ctype.h>

#define SIZE 20

int strcompcase(char\*, char\*);

int main()

{

char \*str1 = NULL;

char \*str2 = NULL;

int n;

str1 = (char\*)malloc(SIZE - sizeof(char));

str2 = (char\*)malloc(SIZE - sizeof(char));

if (NULL == str1 && NULL == str2) {

printf("Malloc failed!!\n");

exit(0);

}

printf("Enter the first string: ");

if(NULL == (fgets(str1, SIZE, stdin))) {

printf("ERROR!!!");

}

printf("Enter the second string: ");

if(NULL == (fgets(str2, SIZE, stdin))) {

printf("ERROR!!!");

}

\*(str1+(strlen(str1))-1) = '\0';

\*(str2+(strlen(str2))-1) = '\0';

printf("%d\n",strcompcase(str1,str2));

free(str1);

free(str2);

str1 = NULL;

str2 = NULL;

return 0;

}

int strcompcase(char \*str1,char \*str2)

{

while(\*str1 != '\0' && \*str2!= '\0') {

\*str1 = tolower(\*str1);

\*str2 = tolower(\*str2);

if(\*str1 == \*str2 && (strlen(str1) == strlen(str2))) {

\*str1++;

\*str2++;

return 0;

} else if(\*str1 == \*str2 && (strlen(str1) > strlen(str2))) {

return 1;

} else if(\*str1 == \*str2 && (strlen(str1) < strlen(str2))) {

return -1;

} else {

printf("String are not equal\n");

return 2;

break;

}

}

}

* **String Copy**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define SIZE 20

char \*strcopy(char\*, char\*);

int main()

{

char \*src = NULL;

char \*dest = NULL;

src = (char\*)malloc(SIZE \* sizeof(char));

dest = (char\*)malloc(SIZE \* sizeof(char));

if (NULL == src && NULL == dest) {

printf("Malloc failed!!\n");

exit(0);

}

printf("Enter the string: ");

if(NULL == (fgets(src, SIZE, stdin))) {

printf("ERROR!!!");

}

\*(src + (strlen(src)) - 1) = '\0';

printf("string is %s\n", strcopy(dest, src));

free(src);

free(dest);

src = NULL;

dest = NULL;

return 0;

}

char \*strcopy(char \*dest,char \*src)

{

char \*temp = dest;

while(\*src) {

\*dest = \*src;

\*dest++;

\*src++;

}

\*dest = '\0';

return temp;

}

* **String Index**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define SIZE 20

int strindex(char\*, char);

int main()

{

char \*str = NULL;

char ch;

str = (char\*)malloc(SIZE - sizeof(char));

if (NULL == str) {

printf("Malloc failed!!\n");

exit(0);

}

printf("Enter the first string: ");

if(NULL == (fgets(str, SIZE, stdin))) {

printf("ERROR!!!");

}

\*(str+(strlen(str))-1) = '\0';

printf("Enter the character to search: ");

scanf("%c", &ch);

printf("Index: %d\n", strindex(str, ch));

free(str);

str = NULL;

return 0;

}

int strindex(char \*str, char ch)

{

int index = 0;

while (\*str) {

if (\*str == ch) {

return index;

break;

}

index++;

str++;

}

printf("Character not found.\n");

}

* **Insert character in a String**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <ctype.h>

#define SIZE 50

char \*strinschr(char\*, const char, int);

int main()

{

char \*str = NULL;

char c;

int pos;

str = (char\*) malloc (SIZE - sizeof(char));

if (NULL == str) {

printf("Malloc failed!!\n");

exit(0);

}

printf("Enter the first string: ");

if(NULL == (fgets(str, SIZE, stdin))) {

printf("ERROR!!!");

}

\*(str + (strlen(str)) - 1) = '\0';

printf("Enter the character to be inserted in string:");

scanf("%c", &c);

printf("Enter the position of character in string:");

scanf("%d", &pos);

printf("String is %s\n",strinschr(str, c, pos));

free(str);

str = NULL;

return 0;

}

char \*strinschr(char \*str, const char c, int pos)

{

char \*temp = str;

int i = strlen(str);

while (\*str++) {

}

while (i > 0) {

\*str = \*(str - 1);

if (i == pos) {

\*(str - 1) = c;

break;

}

str--;

i--;

}

return temp;

}

**String nappend**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define SIZE 20

char \*strnappend(char\*, char\*,int);

int main()

{

char \*src = NULL;

char \*dest = NULL;

int n;

src = (char\*) malloc (SIZE - sizeof(char));

dest = (char\*) malloc (SIZE - sizeof(char));

if (NULL == src && NULL == dest) {

printf("Malloc failed!!\n");

exit(0);

}

printf("Enter the first string: ");

if(NULL == (fgets(src, SIZE, stdin))) {

printf("ERROR!!!");

}

printf("Enter the second string: ");

if(NULL == (fgets(dest, SIZE, stdin))) {

printf("ERROR!!!");

}

\*(src + (strlen(src)) - 1) = '\0';

\*(dest + (strlen(dest)) - 1) = '\0';

printf("Enter the size :");

scanf("%d",&n);

strnappend(src, dest,n);

printf("string is :%s\n",src);

free(src);

free(dest);

src = NULL;

dest = NULL;

return 0;

}

char \*strnappend(char \*src,char \*dest, int n)

{

int i = 1;

while(\*src != '\0') {

src++;

}

while(i <= n) {

\*src = \*dest ;

src++;

dest++;

i++;

}

\*src == '\0';

return src;

}

**String ncopy**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define SIZE 20

char \*strncopy(char\*, char\*, int);

int main()

{

int n;

char \*src = NULL;

char \*dest = NULL;

src = (char\*)malloc(SIZE - sizeof(char));

dest = (char\*)malloc(SIZE - sizeof(char));

if (NULL == src && NULL == dest) {

printf("Malloc failed!!\n");

exit(0);

}

printf("Enter the string: ");

if(NULL == (fgets(src, SIZE, stdin))) {

printf("ERROR!!!");

}

printf("Enter size of the string to be copied :");

scanf("%d", &n);

\*(src+(strlen(src))-1) = '\0';

strncopy(dest, src, n);

printf("string is %s\n", dest);

free(src);

free(dest);

src = NULL;

dest = NULL;

return 0;

}

char \*strncopy(char \*dest,char \*src,int n)

{

int i = 1;

while(i <= n) {

\*dest = \*src;

dest++;

src++;

i++;

}

\*dest = '\0';

return dest;

}

* **String Palindrome**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define SIZE 20

int strpalin(char\*);

int main()

{

char \*str = NULL;

str = (char\*) malloc (SIZE - sizeof(char));

if(NULL == str) {

printf("Malloc failed!!");

exit(0);

}

printf("Enter the string: ");

if(NULL == (fgets(str, SIZE, stdin))) {

printf("fgets failed for str!!");

}

\*(str + (strlen(str)) - 1) = '\0';

if (strpalin(str)) {

printf("String is Palindrome\n");

} else {

printf("String is not a Palindrome\n");

}

free(str);

str = NULL;

return 0;

}

int strpalin(char \*str)

{

char \*len = str + strlen(str) - 1;

while (len > str) {

if (\*str != \*len) {

return 0;

break;

}

\*str++;

\*len--;

}

return 1;

}

* **String Reverse**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define SIZE 20

char\* strrev(char\*);

int main()

{

char \*str = NULL;

str = (char\*) malloc (SIZE - sizeof(char));

printf("Enter the string: ");

if(NULL == str) {

printf("Malloc failed!!");

exit(0);

}

if(NULL == (fgets(str, SIZE, stdin))) {

printf("ERROR!!!");

}

\*(str + (strlen(str)) - 1) = '\0';

printf("Reverese of string is %s\n", strrev(str));

free(str);

str = NULL;

return 0;

}

char\* strrev(char \*str)

{

char \*temp1 = str;

char temp;

char \*len = (str + (strlen(str) - 1));

while (len > str) {

temp = \*str;

\*str = \*len;

\*len = temp;

str++;

len--;

}

return temp1;

}

* **String span**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define SIZE 20

int strspan(char\*, char\*);

int main()

{

char \*str1 = NULL;

char \*str2 = NULL;

str1 = (char\*)malloc(SIZE - sizeof(char));

str2 = (char\*)malloc(SIZE - sizeof(char));

if (NULL == str1 && NULL == str2) {

printf("Malloc failed!!\n");

exit(0);

}

printf("Enter the first string: ");

if(NULL == (fgets(str1, SIZE, stdin))) {

printf("ERROR!!!");

}

printf("Enter the second string: ");

if(NULL == (fgets(str2, SIZE, stdin))) {

printf("ERROR!!!");

}

\*(str1 + (strlen(str1)) - 1) = '\0';

\*(str2 + (strlen(str2)) - 1) = '\0';

printf("Count is: %d\n", strspan(str1, str2));

free(str1);

free(str2);

str1 = NULL;

str2 = NULL;

return 0;

}

int strspan(char \*str1, char \*str2)

{

int count = 0;

int i = 0;

char \*temp = str2;

while (\*str1) {

while (\*str2) {

if (\*str1 == \*str2) {

count++;

break;

}

\*str2++;

}

i++;

if (count < i) {

return count;

break;

}

str2 = temp;

\*str1++;

}

}

* **String Squeeze**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define SIZE 20

char\* strsqueeze(char\*);

int main()

{

char \*str = NULL;

str = (char\*) malloc (SIZE - sizeof(char));

if(NULL == str) {

printf("Malloc failed!!");

exit(0);

}

printf("Enter the string: ");

if(NULL == (fgets(str, SIZE, stdin))) {

printf("fgets failed for str!!");

}

\*(str + (strlen(str)) - 1) = '\0';

printf("Squeezed string is: %s\n", strsqueeze(str));

free(str);

str = NULL;

return 0;

}

char\* strsqueeze(char \*str)

{

char \*temp1;

char \*temp = str;

while (\*str) {

if (\*str == \*(str + 1)) {

temp1 = str;

while (\*str) {

\*str = \*(str + 1);

\*str++;

}

str = temp1;

} else {

\*str++;

}

}

return temp;

}

* **String Token**

#include <stdio.h>

#include <stdlib.h>

#define SIZE 20

char\* mystrtok(char \*str, char \*delim);

int main()

{

char \*str = NULL;

char \*delim = NULL;

str = (char\*) malloc (SIZE \* sizeof(char));

if (NULL == str) {

printf("Malloc failed\n");

exit(1);

}

printf("Enter the string: \n");

if (NULL == fgets(str, SIZE, stdin)) {

printf("Error in fgets\n");

}

delim = (char\*) malloc (SIZE \* sizeof(char));

if (NULL == delim) {

printf("Malloc failed\n");

exit(1);

}

printf("Enter the delimeter: \n");

if (NULL == fgets(delim, SIZE, stdin)) {

printf("Error in fgets\n");

}

printf("%s\n", mystrtok(str, delim));

printf("%s\n", mystrtok(NULL, delim));

printf("%s\n", mystrtok(NULL, delim));

printf("%s\n", mystrtok(NULL, delim));

}

char\* mystrtok(char \*str, char \*delim)

{

static char \*index;

if(str != NULL) {

index = str;

} else {

str = index;

}

if(\*index == '\0') {

return NULL;

}

while(\*index != '\0') {

for(int i = 0; delim[i] != '\0'; i++) {

if(\*index == delim[i]) {

if(index == str) {

index++;

str++;

} else {

\*index = '\0';

break;

}

}

}

if (\*index == '\0') {

index-0000000000000000000000/ex++;

}

return str;

}

* **String sort**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define SIZE 20

#if 1

char\*\* sort(char\*\* ptr);

int main()

{

int i;

char \*\*ptr = ((char\*\*) malloc (5 \* sizeof(char\*)));

for (i = 0; i < 5; i++) {

ptr[i] = ((char\*) malloc (SIZE \* sizeof(char)));

}

printf("Enter the names: \n");

for (i = 0; i < 5; i++) {

fgets(ptr[i], SIZE, stdin);

\*(ptr[i] + strlen(ptr[i]) - 1) = '\0';

}

for (i = 0; i < 5; i++) {

ptr[i] = ((char\*) realloc (ptr[i], sizeof(ptr[i])));

}

ptr = sort(ptr);

printf("\nSorted names are: \n");

for (i = 0; i < 5; i++) {

printf(" %s ", ptr[i]);

}

printf("\n");

free(ptr);

return 0;

}

char\*\* sort(char\*\* ptr)

{

int i, j;

char \*temp;

for (i = 0; i < 4; i++) {

for (j = i + 1; j < 5; j++) {

if (\*ptr[i] > \*ptr[j]) {

temp = ptr[i];

ptr[i] = ptr[j];

ptr[j] = temp;

}

}

}

return ptr;

}

**Stringremove()**

#include<stdio.h>  
#include<string.h>  
#include<stdlib.h>  
#define SIZE 100

char \*mystrrem(char\*, char\*);  
char \*mystrcpy(char\* , char\*);

int main ()  
{  
        char \*str1 = NULL;  
        str1 = (char\*) malloc(SIZE \* sizeof(char));

        char \*str2 = NULL;  
        str2 = (char\*) malloc(SIZE \* sizeof(char));  
        if(NULL == str1  || NULL == str2) {  
                printf("error memory is not allocated-\n");  
                exit(0);  
        }  
        printf("Enter the string in str1-\n");  
        fgets(str1, SIZE, stdin);  
        \*(str1 + (strlen(str1) - 1)) = '\0';  
        if(NULL == str1) {  
                printf("error fgets is NULL-\n");  
        }

        printf("Enter the string to remove form above string-\n");  
        fgets(str2, SIZE, stdin);  
        \*(str2 + (strlen(str2) - 1)) = '\0';  
        if(NULL == str1) {  
                printf("error fgets is NULL-\n");  
        }

        printf("string - %s\n", mystrrem(str1, str2));  
        free(str1);  
        free(str2);  
        str1 = NULL;  
        str2 = NULL;  
        return 0;

}

char \*mystrrem(char\* str, char\* substr)  
{  
        int result;  
        int count = 0;  
        int j = 0; //  
        char\* strtemp1 = str;

        while(\*strtemp1 != '\0') {  
                if(\*strtemp1 == \*substr) {  
                //      substr1 = substr;  
                        while(\*substr != '\0') {  
                                if(\*strtemp1 == \*substr) {  
                                        strtemp1++;  
                                        substr++;  
                                        j = 1;  
                                        result = 1;  
                                } else {  
                                        j = 1;  
                                        result = 0;  
                                        break;  
                                }  
                        }  
                } else if(j == 1) {  
                        break;  
                } else {  
                        result = 0;  
                        strtemp1++;  
                        count++;  
                }  
        }  
        if(result) {  
                char\* cpypos = str + count;  
                mystrcpy(cpypos,strtemp1);  
        }  
        return str;  
}

char \*mystrcpy(char\* dest, char\* sour)  
{  
        char \*temp = NULL;  
        temp = dest;  
        while(\*temp++ = \*sour++);  
        return dest;  
}

**Memcopy()**

#include <stdio.h>  
#include <string.h>

void\* my\_memcpy(void\*, void\*, int );  
int main ()  
{  
        int a = 18;  
    char \*b;  
        //int \*res = my\_memcpy(b, &a, 13);  
        my\_memcpy(b, &a, 13);  
        printf("%d", \*b);  
        //printf("%d", \*res);  
//      int \*res =(int \*) memcpy(b, &a, 13);  
//      printf("%d", \*b);  
//      printf("%d", \*res);  
}

void\* my\_memcpy(void\* dest, void\* src, int num )  
{  
        char \*temp1 =(char\* ) dest;  
        char \*temp2 =(char\* ) src;  
        while(num) {  
                \*temp1 = \*temp2;  
                temp1++;  
                temp2++;  
                num--;  
        }

        return dest;  
}

**Memmove()**

include <stdio.h>  
#include <string.h>  
#include <stdlib.h>

void\* my\_memmove(void\*, void\*, int );

int main ()  
{  
        int a = 18;  
    char \*b;  
        my\_memmove(b, &a, 13);  
        printf("%d", \*b);  
}

void\* my\_memmove(void\* dest, void\* src, int num )  
{  
        int num1 = num;  
        char \*temp1 =(char\* ) dest;  
        char \*temp2 =(char\* ) src;

        char \*arrtemp = (char \*) calloc(num , sizeof(char));  
        char \*arr = arrtemp;  
        while(num1) {  
                \*arr = \*temp2;  
                arr++;  
                temp2++;  
                num1--;  
        }  
        arr = arrtemp;  
        while(num) {  
                \*temp1 = \*arr;  
                arr++;  
                temp1++;  
                num--;  
        }  
        free(arrtemp);  
        arrtemp = NULL;  
        arr = NULL;  
        temp1 = NULL;  
        temp2 = NULL;  
        return dest;  
}