## Pointer Arithmetic Lab Assignments

**implement below functions using pointer increment or decrement method, use below declarations. Change function name.**

**1) int strlen(char s[]);**

```
#include<stdio.h>
int myStrlen(char s[]){
   char *p=s;
   while(*p){
      p++;
   }
   return p-s;
}

int main(){
   char str[]="anil kumar";
   int length = myStrlen(str);
   printf("length of %s is %d",str,length);
}
```

Output:

Length of  anil kumar is 10

**2) char * strrev( char s[]); // returns string base address**

```
#include <stdio.h>

char *my_strrev(char s[]) {
   char *start = s;
   char *end = s;

   while (*end) {
      end++;
   }
```

```
    end--;

    while (start < end) {
        char temp = *start;
        *start = *end;
        *end = temp;
        start++;
        end--;
    }

    return s;
}

int main() {
    char str[] = "Anil";
    printf("Original: %s\n", str);
    my_strrev(str);
    printf("Reversed: %s\n", str);
    return 0;
}
```

Output:

Original: Anil

Reversed: linA

**3) char * strcpy( char d[] , char s[]);  // returns destination base address**

```
#include<stdio.h>
char *myStrcpy(char d[],char s[]){
    char *dest=d;
    while((*d++=*s++));
    return dest;
}
```

```
int main(){
   char src[]="poter copy Anil";
   char dest[50];
   myStrcpy(dest,src);
   printf("source      : %s\n",src);
   printf("destination  : %s",dest);
}
```

Output:

```
source        : poter copy Anil
destination  : poter copy Anil
```

**4) int  strcmp (char s1[], char s2[]);**

```
#include<stdio.h>
char *myStrcmp(char s1[],char s2[]){
   while(*s1 && (*s1==*s2)){
      s1++;
      s2++;
   }
   return *(unsigned char *)s1-*(unsigned char*)s2;
}
int main(){
   char a[]="anil kumar";
   char b[]="aziz";
   char c[]="bhanu";
   printf("cmp of a and b is:%d\n",myStrcmp(a,b));
   printf("cmp of b and c is:%d\n",myStrcmp(b,c));
   printf("cmp of c and a is:%d\n",myStrcmp(c,a));
}
```

Output:

```
cmp of a and b is:-12
cmp of b and c is:-1
cmp of c and a is:1
```

**5) int strncmp(char s1[]. char s2[], int n);**

```
#include <stdio.h>

int my_strncmp(char s1[], char s2[], int n) {
   while (n > 0 && *s1 && (*s1 == *s2)) {
      s1++;
      s2++;
      n--;
   }

   if (n == 0) {
      return 0;
   }

   return *(unsigned char *)s1 - *(unsigned char *)s2;
}

int main() {
   char a[] = "pointer";
   char b[] = "pointed";

   printf("Compare first 5 chars: %d\n", my_strncmp(a, b, 5));
   printf("Compare first 7 chars: %d\n", my_strncmp(a, b, 7));
   printf("Compare first 3 chars: %d\n", my_strncmp(a, b, 3));

   return 0;
}
```

Output:

```
Compare first 5 chars: 0
Compare first 7 chars: 14
Compare first 3 chars: 0
```

**6) int stricmp (char s1[], char s2[]);**

```
#include<stdio.h>
char toLower(char c){
   if(c=='A' && c=='Z'){
      return c+32;
   }
   return c;
}
char myStricmp(char s1[],char s2[]){
   while(*s1 && *s2){
   char c1=toLower(*s1);
   char c2=toLower(*s2);
   if(c1!=c2){
      return (unsigned char)c1-(unsigned char)c2;
   }
   s1++;
   s2++;

   }
   return *(unsigned char*)s1-*(unsigned char*)s2;
}
int main(){
   char a[]="ANIL";
   char b[]="anil";
   printf("compare a and b:%d\n",myStricmp(a,b));
   return 0;
}
```

Output:

```
compare a and b:-32
```

**7) char * strcat(char d[], char s[]); // returns destination base address**

```
#include<stdio.h>
int myStrcat(char d[],char s[]){
   char *dest=d;
   while(*dest){
      dest++;
   }
   while(*s){
      *dest=*s;
      dest++;
      s++;
   }
   *dest='\0';
   return d;
}
int main(){
   char d[]="hello";
   char s[]="world";
   myStrcat(d,s);
   printf("combination : %s\n",d);
   return 0;
}
```

Output:

```
combination : helloworld
```

**8) char * strlwr(char s[]); // returns s base address**

```
#include <stdio.h>
char* my_strlwr(char s[]) {
   char *p = s;

   while (*p) {
      if (*p >= 'A' && *p <= 'Z') {
```

```
            *p = *p + 32;
        }
        p++;
    }

    return s;
}

int main() {
    char str[] = "HeLLo WoRLd! 123";

    my_strlwr(str);

    printf("Lowercase string: %s\n", str);
    return 0;
}
```

Output:

Lowercase string: hello world! 123

**9) char * strchr( char s[], char c); // returns address of given character first occurrence in given string**

```
#include<stdio.h>
char *myStrchr(char s[],char c){
    while(*s){
        if(*s==c){
            return s;
        }
        s++;
    }
    if(c=='\0'){
        return s;
    }
```

```
    return NULL;
}
int main(){
    char str[]="pointer Arithmetic";
    char ch='A';
    char *result=myStrchr(str,ch);
    if(result!=NULL){
        printf("character %c is found at positioon:%ld\n",ch,result-str);
        printf("Substring from found character: %s\n", result);
    }
    else{
        printf("charecter is not found");
    }
    return 0;

}
```

Output:

```
character A is found at positioon:8
Substring from found character: Arithmetic
#include<stdio.h>
char *myStrstr(char str[],char sub[]){
    char *s=str;
    if(*sub=='\0'){
        return str;
    }
    while(*s){
        char *p1=s;
        char *p2=sub;
    while(*p1 && *p2 && (*p1==*p2)){
        p1++;
        p2++;
```

```
    }
    if(*p2=='\0'){
        return s;
    }
    s++;
    }
    return NULL;
}
int main(){
    char str[]="pointer arithmetic is poewerfull";
    char sub[]="arithmetic";
    char *result=myStrstr(str,sub);
    if(result!=NULL){
        printf("Substring found at position: %ld\n", result - str);
        printf("Substring in string: %s\n", result);
    }
    else{
        printf("sub string is not found");
    }
    return 0;
}
```

Output:

Substring found at position: 8

Substring in string: arithmetic is poewerfull