### int mStrlen(const char* str)

Input : 문자열

Output: 문자열의 길이

Function : 문자열의 길이를 계산하여 반환한다.

```
while(c!='₩0'){
      c= *(str+cnt);
      cnt++;
```

### char * mStrcpy(const char *str1,char *str2)

Input : 2개의 문자열

Output : 복사된 문자열의 시작주소

Function : 첫번째 문자열을 두번째 문자열을 복제한다.

```
for(i=0;i<l;i++)
        *(str2+i)=*(str1+i);
  *(str2+i)='₩0';
```

### char * mStrncpy(const char *str1,char *str2,int n)

Input : 2개의 문자열과 숫자

Output : 복사된 문자열의 시작주소

Function : 첫번째 문자열을 두번째 문자열에 숫자만큼 복제한다.

```
for(i=0;i<n-1;i++)
```

### char * mStrcat(char *str1,const char *str2)

Input : 2개의 문자열

Output : 결합된 문자열의 시작주소

Function : 첫번째 문자열에 두번째 문자열을 결합한다.

```
for(i=l1;i<(l1+l2);i++)
        *(str1+i)=*(str2+j++);
```

### int mStrcmp(const char *str1,const char *str2)

```
   int l1,l2,i;


   l1=mStrlen(str1);
   l2=mStrlen(str2);


   for(i=0;i<l1 && i<l2;i++){
      if(*(str1+i) != *(str2+i))
         break;
      }
   if(*(str1+i) > *(str2+i))
      return -1;
   else if(*(str1+i) < *(str2+i))
      return 1;
   else if(l1==l2)
      return 0;
   else if(l1>l2)
      return 1;
   else
      return -1;
}        // compare string by dic order
```

Input : 2개의 문자열

Output : 비교 결과

Function : 2개의 문자열을 ASCII 기준으로 비교한다.

### const char* mStrchr(const char *str,char c)

Input : 문자열과 문자

Output : 위치의 주소를 반환

Function : 입력받은 문자를 문자열에서 찾는다.

```
for(int i=0;i<l;i++)
      if(*(str+i)==c)
```