```
int num = 10;
```

```
const int *ptr = &num;
```

- ptr is not constant
- num is constant
- num++ ✓
- *ptr = 3 ✗

```
int *const ptr = &num;
```

- ptr is constant
- num is not constant
- num++ ✓
- *ptr = 3 ✓
- ptr++ ✗

```
const int num = 10;
```

```
const int *ptr = &num
```

- ptr is not constant
- num is constant
- num++ ✗
- *ptr = 3 ✗

```
int *const ptr = &num;
```

- ptr is constant
- num is not constant
- num++ ✗
- *ptr = 3 ✓
- ptr++ ✗

```
int num=10;
```

```
const int *const ptr = &num;
```

- ptr is constant
- num is constant
- num ++ ✓
- *ptr = 2. ✗
- ptr++ ✗

```
const int num=10;
```

```
const int *const ptr = &num;
```

- ptr is constant
- num is constant
- num++ ✗
- *ptr = 2. ✗
- ptr++ ✗

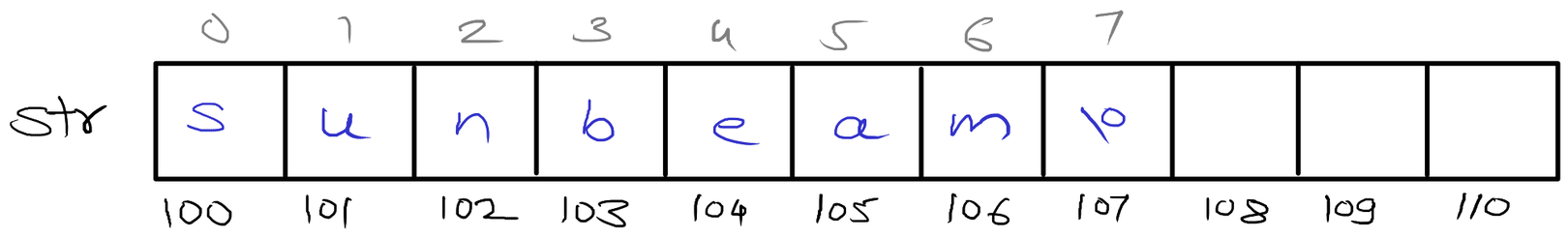
```
int *ptr = &num;
```

- ptr is not constant
- num is not constant
- *ptr = 2. ✓
- ptr++ ✓
- num ++

```
int *ptr = &num;
```

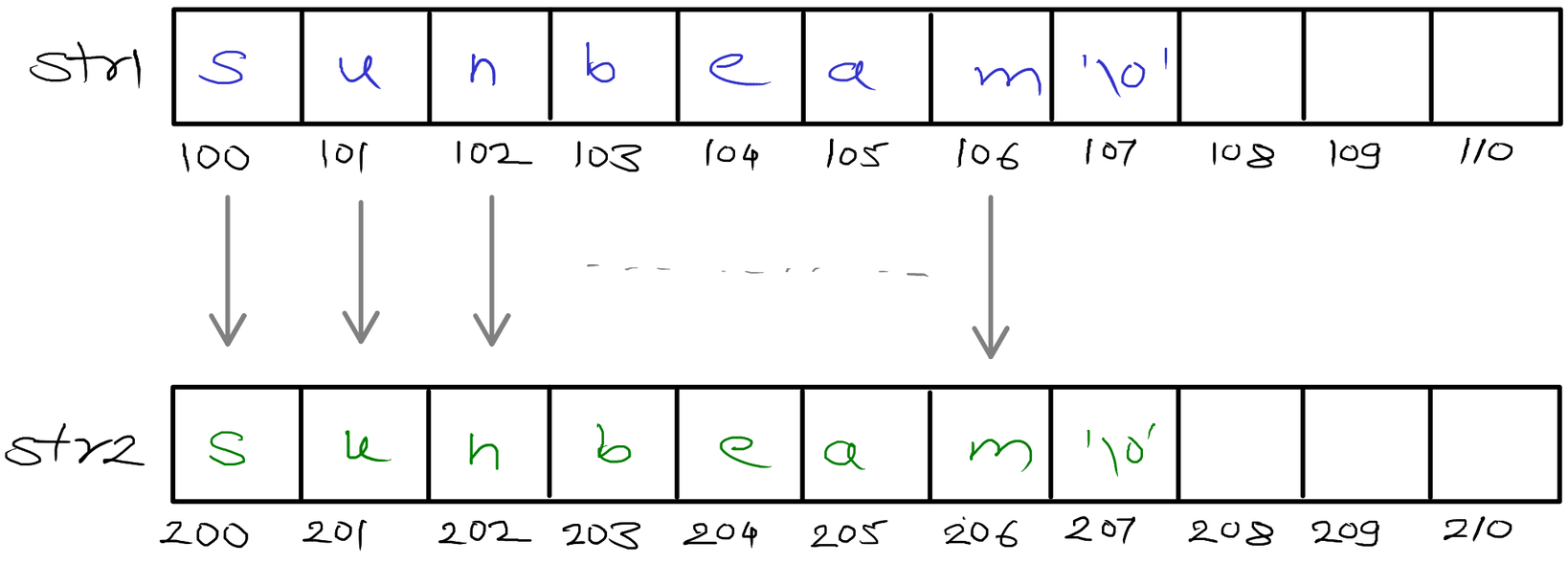
- ptr is not constant
- num is not constant
- *ptr = 2 ✓
- ptr++ ✓
- num++ ✗

strlen




```
size_t mystrlen(const char *s)
{
    size_t length=0;
    while(*s!='\0')
    {
        length++;
        s++;
    }
    return length;
}
```

strcpy

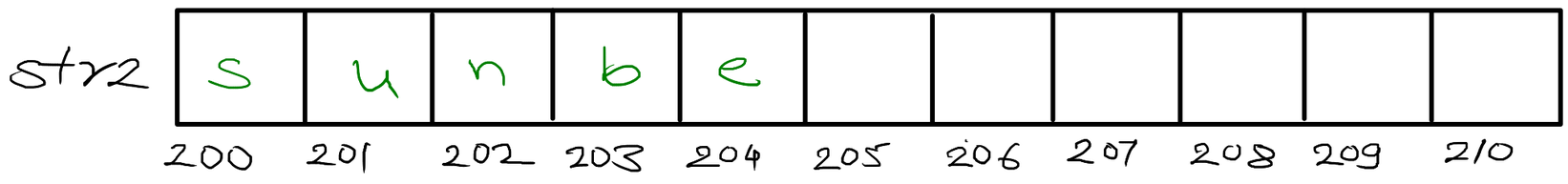


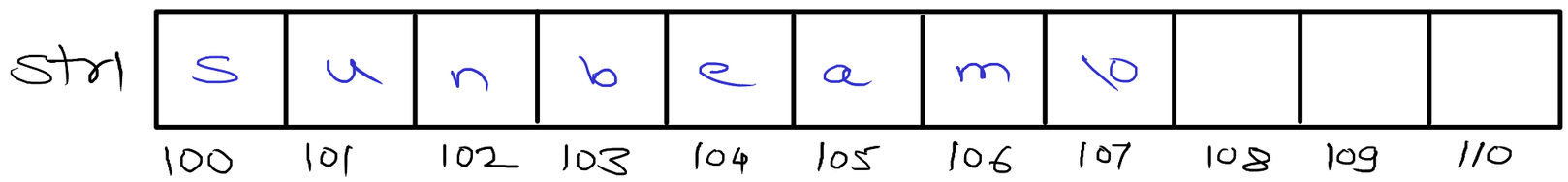

```
char *my_strcpy(char *dest, const char *src)
{
        int i;
        for(    i=0; src[i] != '\0', i++)
                dest[i] = src[i];

        dest[i]='\0';
        return dest;
}
```

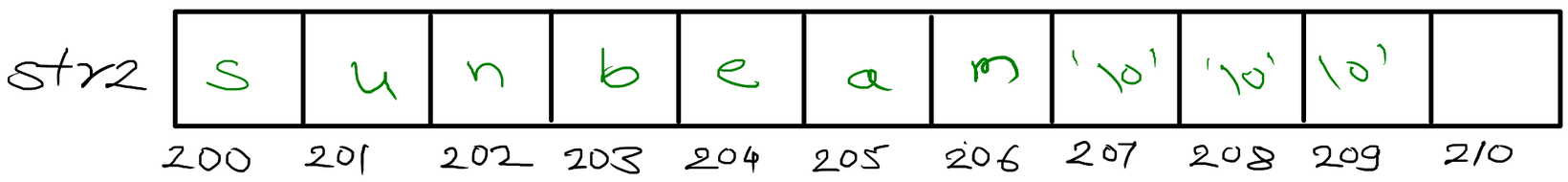
strncpy

strncpy(str2, str1, 5);

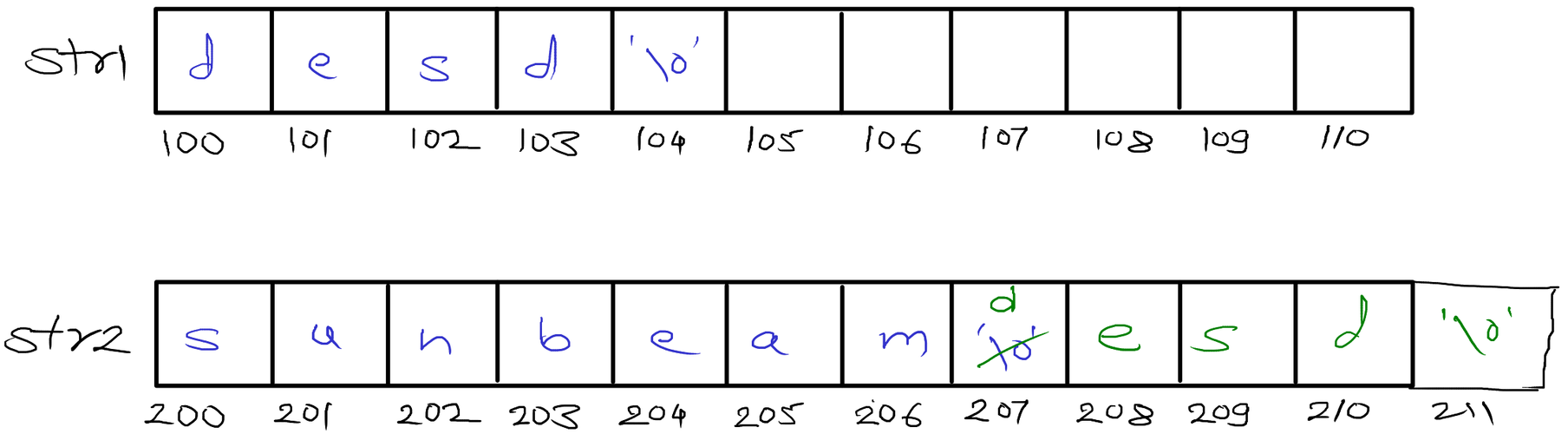




strncpy(str2, str1, 10);

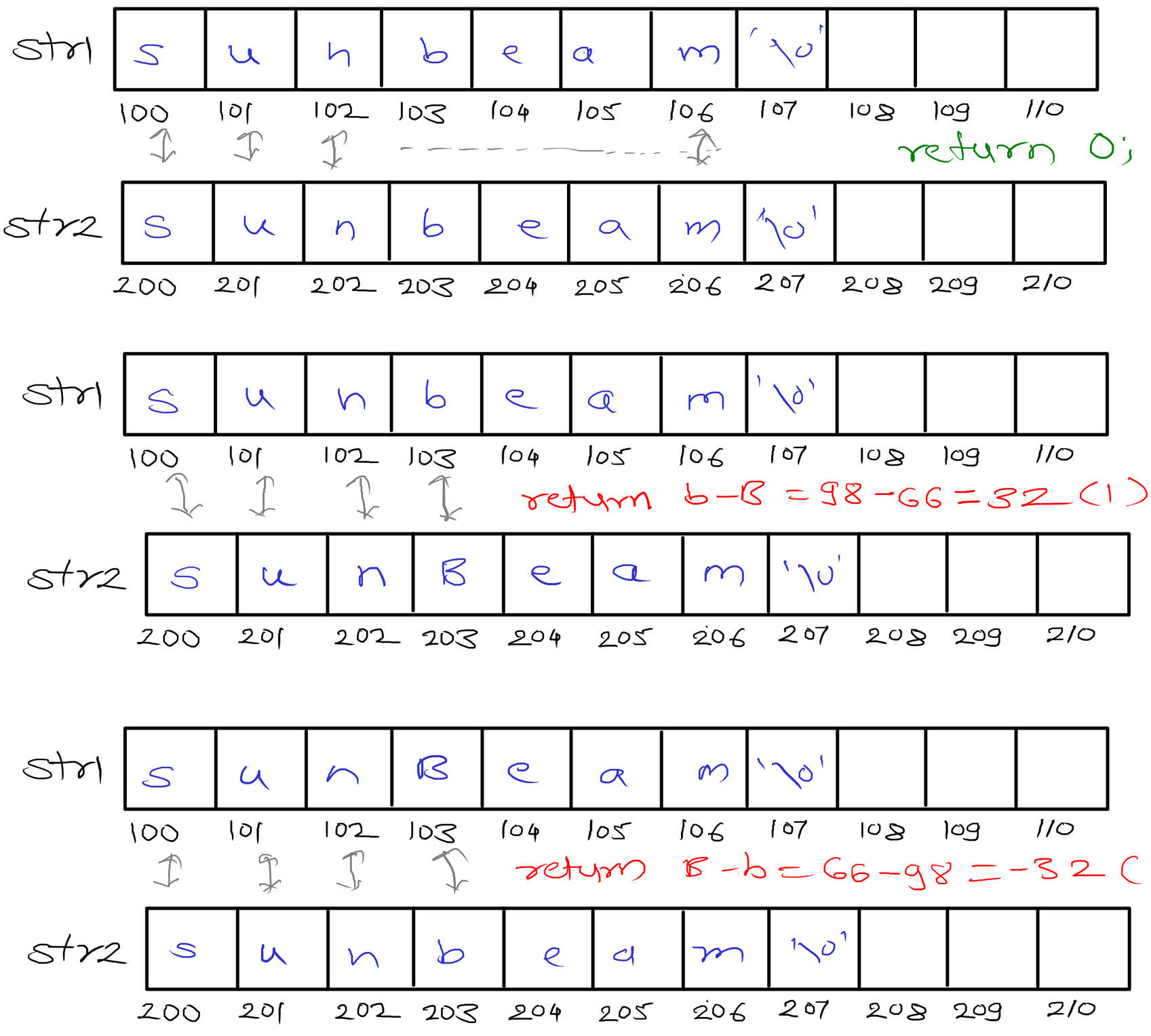


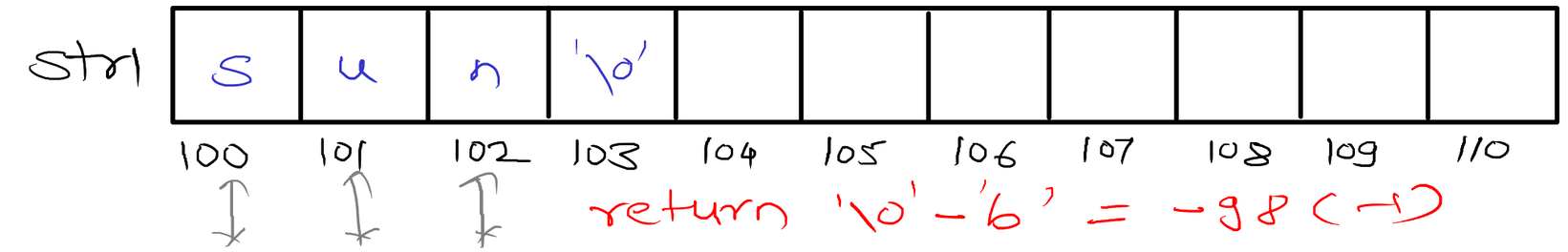
strcat

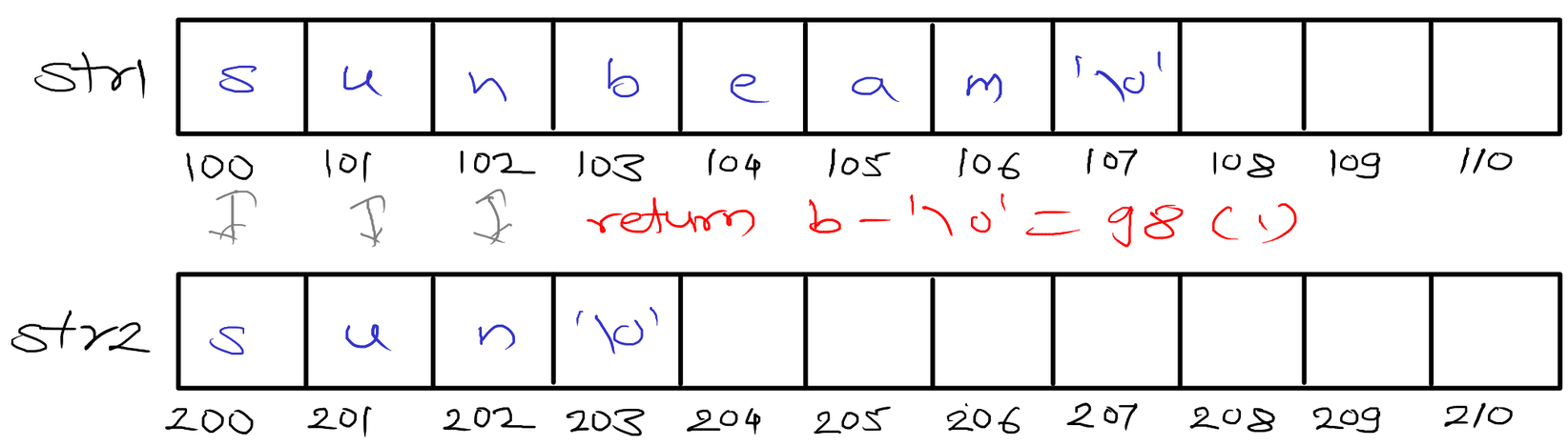



```
char *my-strcat( char *dest, const char *src)
{
    int i=0;
    for( ; dest[i] != '\0'; i++);
    while(*src != '\0')
    {
        dest[i++]= *src;
        src++;
    }
    dest[i] = '\0';
    return dest;
}
```

strcmp


str1
s u h b e a m '\0'
100 101 102 103 104 105 106 107 108 109 110
return 0;
str2
s u n b e a m '\0'
200 201 202 203 204 205 206 207 208 209 210
str1
s u n b e a m '\0'
100 101 102 103 104 105 106 107 108 109 110
return b-B = 98-66 = 32 (1)
str2
s u n B e a m '\0'
200 201 202 203 204 205 206 207 208 209 210
str1
s u n B e a m '\0'
100 101 102 103 104 105 106 107 108 109 110
return B-b = 66-98 = -32 (-1)
str2
s u n b e a m '\0'
200 201 202 203 204 205 206 207 208 209 210


str1
s
u
n
'\0'
100
101
102
103
104
105
106
107
108
109
110
return '\0' - 'b' = -98 (-)
str2
s
u
n
b
e
a
m
'\0'
200
201
202
203
204
205
206
207
208
209
210

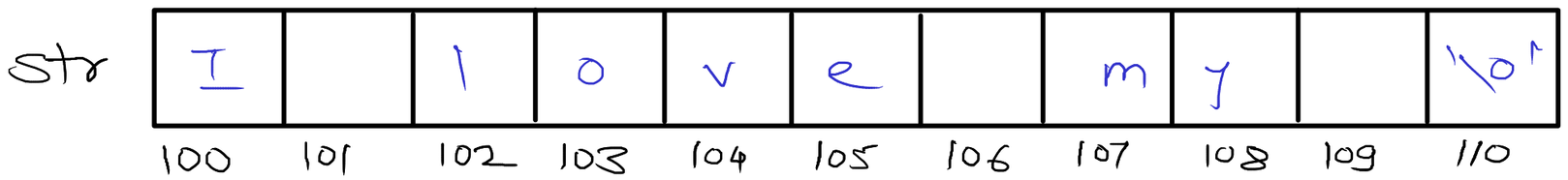

str1
s
u
n
b
e
a
m
'\0'
100
101
102
103
104
105
106
107
108
109
110
return b - '\0' = 98 (+)
str2
s
u
n
'\0'
200
201
202
203
204
205
206
207
208
209
210

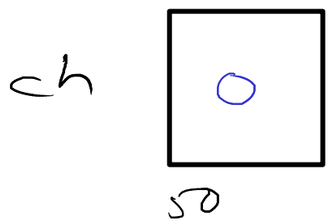
```
int my-strcmp(const char *s1, const char *s2)
{
    int i=0;
    for( ; s1[i] != '\0'; i++)
    {
        if(s1[i] != s2[i])
            return s1[i]-s2[i];

    }
     return s1[i]-s2[i];
}
```

```
int my-strcmp(const char *s1, const char *s2)
{
    while(*s1 != '\0')
    {
        if(*s1 != *s2)
            return *s1 - *s2;

        s1++;
        s2++;
    }
    return *s1 - *s2;
}
```






```
char *my_strchr(const char *s, int c)
{
    for(i=0; s[i] != '\0'; i++)
    {
        if(s[i] == c)
            return &s[i];
    }
    if(c == '\0')
        return &s[i];
    return NULL;
}
```

```
char *my_strstr(const char *string, const char *substring)
{
        size_t len = strlen(substring);
        for(int i=0; string[i] != '\0'; i++)
        {
                if(string[i] == substring[0])
                        if(!strncmp(string+i, substring, len))
                                return string+i;

        }
        return NULL;
}
```

i

string = i am desd student

substring = desd length = 4