13/04/21

Strings

Date
Page

ASCII codes (for only English language)

A - 65	a = 97	0 = 48
B - 66	b = 98	1 = 49
!	!	!
Z = 90	z = 122	9 = 57

↵ = 10 , space = 13 , esc = 27

* ASCII code stores 1 byte in computer memory.
* ASCII codes Ranges from 0 to 127. (Total - 128)

UNICODES (Subset of ASCII)

↓

It works for all language

→ takes memory - 2 bytes (16 bits)

→ These are represented in Hexadecimal form & in four digits.

Char temp;

✓ temp = 'A';

✗ temp = 'AB';

✗ temp = A;

✗ temp = "A";

temp

A
65

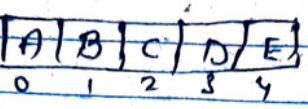
printf ("%d", temp); → 65

printf ("%c", temp); → A

* Array of character -

char x[5];

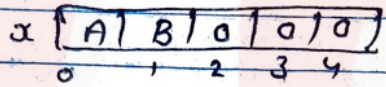
char x[5] = {'A', 'B', 'C', 'D', 'E'};



char x[] = {'A', 'B', 'C', 'D', 'E'};

char x[5] = {65, 66, 67, 68, 69};

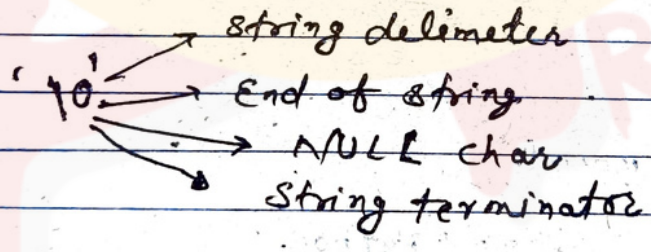
char x[5] = {'A', 'B'};



→ To storing name, word or sentence in computer we use character array i,e string.

→ To show the end of string we computer use '\0'.

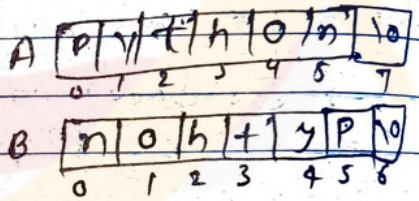
→ Every string terminated with a null ('\0') char.



* Reverse a string

Using another array

```
int main()
{
  char A[] = "Python";
  char B[7];
  int i;
  for(i=0; A[i]!='\0'; i++)
  {
  }
  i = i-1;
  for(j=0; j>=0; i--, j++)
  {
    B[j] = A[i];
  }
  B[j] = '\0';
  printf("%s", B);
}
```



2nd method

```
int main()
{
  char A[] = "Python";
  char t;
  int i, j;
  for(j=0; A[j]!='\0'; j++)
  {
  }
  j = j-1;
  for(i=0; i<j; i++, j--)
```

```
    {
        t = A[i];
        A[i] = A[j];
        A[j] = t;
    }
    printf("%s", A);
}
```

17/04/22

* Comparing String and checking palindrome -

```
int main()
{
    char A[] = "Painter";
    char B[] = "Painting";
    int i, j;
    for (i=0,j=0; A[i]!='\0' && B[j]!='\0'; i++,j++)
    {
        if (A[i]!=B[j])
            break;
    }
    if(A[i]==B[j])
        printf("Equal");
    else if (A[i]<B[j])
        printf("Smaller");
    else
        printf("greater");
```

NOTE- If both strings are not in same case (lower or upper case) then make it in same before comparing.

Date
Page

* Pallindrom -

If reversing of a word also same as given word then it is pallindrom.

word - m a d a m

reverse word - m a d a m

```
int main()
{ char A[] = "Madam"
  char B[6];
```

(i) Reverse

(ii) Compare

2nd-method - Comparing the first letter of the word and last letter of word and simultaneously we can find its palindrom or not.

* Finding Duplicates in a string -

(It will work on only lower case strings)

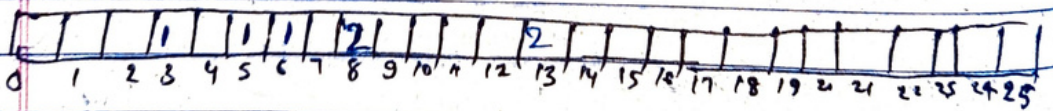
Using hash

ASCII → 102 105 110 100 105 110 103

A | f | i | n | d | i | n | g | \0 |

0 1 2 3 4 5 6 7

H:



f = 102 - 97 = 5

i = 105 - 97 = 8

n = 110 - 97 = 13

d = 100 - 97 = 3

g = 103 - 97 = 6

```
int main()
{ char A[] = "finding"
  int H[26], i;
for (i=0; A[i] != '\0'; i++)
  {
n ← H[A[i]-97] += 1;
  }
for (i=0; i<126; i++)
  if (H[i] > 1)
  { printf("%c", i+97);
n ← printf("%d", H[i]);
  }
}
}
```

time − n + n
= 2n
O(n)