Homework Assignment 15

Q1. State True/False :

i. The sequence **'abcd'** signifies a pointer that contains the address of 'a'
ii. The empty string **" "** uses no memory
iii. The string used as the first argument to **sscanf** changes every time the function reads it
iv. if `p` is declared as `char *p;` and `arr` is declared as `char arr[10];` the statement `arr = p;` is legal
v. if `char *p = "Python";` the statement `p[0] = 'p';` is illegal
vi. A sting s1 cannot be copied to the string s2 using s2 = s1
vii. For storing strings of unequal length, an array of car pointers is a better option than a 2D character array
viii. The command line arguments to a C program are read by the program as strings

Q2. Fill in the blanks.

i. A function looks for the ------ character to determine the end of a string
ii. A string can be declared as a pointer og type ------ or as an array of type --------
iii. The ------- function must not be used because it can overflow the buffer
iv. The -------- function reads data from a string and the ----- function writes data to string
v. The ---- function checks for an alphabetic character, and the ---- function checks for a digit
v. For a program to accept command line argument, the signature of the --- function must be modified.

Q3. Answer the following :

i. Why is it advantageous to use sscanf to read a string instead of using scanf to read standard input?
ii. The function call strcmp("somalia","somalis"); returns ------
iii. Explain why standard technique of passing pointers to a function doesn't work when swapping two strings
iv. `char *p = " Redmi"; and char arr[] = "Redmi";` it is possible to use :
a. `p++` b. `p+1` c. `arr+1`

Q4. Write a program that accepts five lines from the keyboard and saves them in two 2D arrays depending on whether the line length is longer than 20 characters (excluding newline and NULL). The program must print contents of both arrays.

Q5. Give output with proper explanation :

i.
```
int main()
{
      char s[] = "Rendezvous !";
      printf("\n %d" ,*(s + strlen(s)));
}
```

ii.
```
int main()
{
      printf(5+"IIITALLAHABAD);
}
```

iii.
```
int main()
{
      char ch[20];
      int i;
      for( i = 0 ; i&lt;19 ; i++)
            *(ch+i) = 67;
      *(ch+i) = '\0' ;
      printf("%s" , ch):
}
```

Q6. Give an algorithm to reverse a string