String Functions Assignment:

Mandatory

1. WAP to prompt and read a line of text with words separated by space. Tokenise and extract the words. Display them. Impement the following functions for this.
    a. int tokenise(char *arr); //tokenise and display tokens , return number of tokens to the caller

    [Use strtok() to tokenise]

```c
#include <stdio.h>
#include<string.h>
#include <stdlib.h>
#define MAX 50
int tokenise(char *);
int main() {
    char str[MAX];
    char arr[20];
    int i=0,c;
    fgets(str,MAX,stdin);
    str[strlen(str)-1]='\0';
    c=tokenise(str);
    printf("count :%d\n",c);

    return 0;
}
int tokenise(char *str)
{   int c;
    char *ptr;

    ptr=strtok(str, " ");
    fputs(ptr,stdout);
    c=1;
    printf("\n");
    while((ptr=strtok(NULL," "))!=NULL)
    {
        fputs(ptr,stdout);
      printf("\n");
      c++;
    }

 return c;
}
```

```
hello world welcome
hello
world
welcome
count :3
```

2. Implement your own strncat() which shall concatenate n characters from src to dest.

```
char *strncat(char *dest, const char *src, size_t n)
```

```c
#include <stdio.h>
char *strncati(char *,  char *,int);
int main() {
    char s[30]="pooja";
    char d[]="bevara";
    char *s1=s;
    char *s2=d;
    printf("hello\n");
    strncati(s1,s2,2);
    printf("cat: %s",s1);
    return 0;
}
char *strncati(char *s1,char *s2,int n)
{
    while(*s1 !='\0')
        s1++;
       int i=0;
    while(i<n)
    {
        *s1=*s2;
        s1++;
        s2++;
        i++;
        }
    s1='\0';

    //printf("%s",s2);
    return s1;
}
```

```
cat: poojabe
```

3. WAP to
    a. Search for and replace the vowel characters (upper and lower case) with “ay” in a word read from user. Consider a maximum word length of 40 characters.
    b. Identify the test inputs to be used
    c. Display updated string

```
Enter a word : Hello
Updated word: Hayllay
```

```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>

#define MAX_LEN 40

void replace_vowels_with_ay(char *str) {
    int i = 0;
    char temp[MAX_LEN * 2];

    int j = 0;
    while (str[i] != '\0') {
        if (str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u' ||
            str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U') {
            temp[j++] = 'a';
            temp[j++] = 'y';
        } else {
            temp[j++] = str[i];
        }
        i++;
    }

    temp[j] = '\0';  // Null-terminate the new string

    strcpy(str, temp);
}

int main() {
    char word[MAX_LEN + 1];

    printf("Enter a word : ");
    fgets(word, MAX_LEN + 1, stdin);

    word[strcspn(word, "\n")] = '\0';

    replace_vowels_with_ay(word);

    printf("Updated word: %s\n", word);

    return 0;
}
```