Assignment No.9

1.//strcat

```
#include<stdio.h>
#include<string.h>
int main()
{
        char str1[]={"Hello"};
        char str2[]={"Student"};
        strcat(str1,str2);
        printf("%s",str1);
}
```

2. //strcpy: copy one string in to another string.

```
#include<stdio.h>
#include<string.h>
int main()
{
        char str1[]={"Ankita"};
        char str2[10];
   strcpy(str2,str1);
        printf("The copied string is: %s\n",str2);


}
```

3. //strcat

```
#include<stdio.h>
#include<string.h>
int main()
{
        char str1[]={"Hello"};
        char str2[]={"Student"};
        strcat(str1,str2);
        printf("%s",str1);
}
```

4. //strcmp

```
#include<stdio.h>
#include<string.h>
void main()
{
        char str1[]={"Hi Firstbit"};
        char str2[]={"Solution"};
        strcmp(str1,str2);

        if(str1>str2)
        {
          printf("The string one is greater than string second\n");
     }
        else if(str1<str2)
        {
          printf("The string one is less than string two\n");
    }
```

```
   else
    {
       printf("The both string are equal");
            }

}
```

5. //strncpy

```
#include<stdio.h>
#include<string.h>
void main()
{
            char str1[]=("student");
            char str2[20];
            strncpy(str2,str1,2);
            str2[2]='\0';
            printf("The copy string is %s \n",str2);
            return 0;
}
```

6. //strncat:concatenates a specified number of characters from one string to another

```
#include<stdio.h>
#include<string.h>
int main()
{
            char str1[20]="Good";
            char str2[]="Student";
```

```
        strncat(str1,str2,4);
        printf("The concatenation string is %s\n",str1);
        return 0;
}
```

7. //strncp:Work like strcmp(),but bonly compares the first n characters.

```
#include<stdio.h>
#include<string.h>
void main()
{
        char str1[]="programming";
        char str2[]="program";
        if(strncmp(str1,str2,7)==0)
         {
                printf("First 7 chracters match\n");
         }
         else
         {
          printf("Dont match the character\n");
    }

}
```

7. //strchr: searches for the first occurence of a specific character in a string.

```
#include<stdio.h>
#include<string.h>
void main()
{
```

```
	char str[]="development";
	char *ptr=strchr(str,'e');
	if(ptr)
	{
		printf("First 'e' found at position:%d\n",ptr-str);
	}
}
```

8. //strchr: searches for the first occurence of a specific character in a string.

```
#include<stdio.h>
#include<string.h>
void main()
{
	char str[]="development";
	char *ptr=strchr(str,'e');
	if(ptr)
	{
		printf("First 'e' found at position:%d\n",ptr-str);
	}
}
```

9. //strrrev: this function reverse the string in c

```
#include<stdio.h>
#include<string.h>
void main()
{

        char str[10]="Firstbit";
        strrev(str);
        printf("The reverse string is :%s\n",str);
}
```

10. //strrchr:searches for the last occurence of a character in a string

```
#include<stdio.h>
#include<string.h>
int main()
{
        char str1[]="firstbit";
        char *ptr=strrchr(str1,'e');
        if(str1)
          printf("Last occurence of 't' found at position:%d\n",ptr-str1);
          return 0;
}
```

11. //strstr:searches for the firdt occurence of a substring in a string

```
#include<stdio.h>
#include<string.h>
int main()
```

```
{
        char text[]="Hello firstbit solution";
        char *sub= strstr(text,"Hello");
        if(sub)
          {
                printf("Substring found at: %s\n",sub);
          }
}
```

12. //strtok: splits a string into tokens based on a delimeter

```
#include<stdio.h>
#include<string.h>
int main()
{
        char str[]="Hi,i,am,ankita";
        char *token=strtok(str,",");

        while(token!='\0')
         {
                printf("%s\n",token);
                token=strtok('\0',",");
         }
         return 0;
}
```

13. //strerror:converts an error code(typically error)into a uman-readable string message.

```
#include<stdio.h>
#include<string.h>
#include<errno.h>
int main()
{
	FILE *f=fopen("nonexistent.txt","r");
	if(!f)
	{
		printf("Error:%s\n",strerror(errno));
	}
	return 0;
}
```

14. //memset:Fills a block of memory with a specific value

```
#include<stdio.h>
#include<string.h>
int main()
{
	char str[20];
	memset(str,'@',10);
	str[10]='\0';
	printf("After memset:%s\n",str);
	return 0;
}
```

16. //memcpy: copies a specified number of bytes from source to destination

```
#include<stdio.h>
#include<string.h>
int main()
{
        char str[20]="firstbit solution";
        char dest[10];
        memcpy(dest,str,strlen(str)+1);
        printf("Copies the byte:%s\n",dest);
        return 0;
}
```

17. //memcmp:compares two locks of bmemory byte by byte for a specified size

```
#include<stdio.h>
#include<string.h>
int main()
{
        char str1[]="Firstbit";
        char str2[]="Solution";
        if(memcmp(str1,str2,3)==0)
          {
                printf("Memorty blocks are equal\n");
          }
        else
         {
                printf("Two blocks are not equal");
         }
          return 0;
```

```
}
```

18. //strdup: allocates memory and create a duplicate of the given string.

```
#include<stdio.h>
#include<string.h>
int main()
{
        char str1[]="C programming";
        char *dubli=strdup(str1);
        printf("Copied String:%s\n",dubli);
        free(dubli);
}
```

19.// toupper:

```
#include<stdio.h>
#include<string.h>
int main()
{
        char ch='b';
        printf("Before conversion %c\n",ch);

        ch=toupper(ch);
        printf("After conversion:%c\n",ch);

}
{

        char ch='B';
        printf("Before conversion %c\n",ch);
```

```
	ch=tolower(ch);
	printf("After conversion:%c\n",ch);

}
```

20. //tolower

```
#include<stdio.h>
#include<string.h>
int main()
{
	char ch='b';
	printf("Before conversion %c\n",ch);

	ch=toupper(ch);
	printf("After conversion:%c\n",ch);

}
{

	char ch='B';
	printf("Before conversion %c\n",ch);

	ch=tolower(ch);
	printf("After conversion:%c\n",ch);

}
```