Logic Building Assignment : 13

Complete below code snippets it contains only service provider function.

Write entry point function to call below helper functions separately.

Create separate visual Studio project for each problem statement separately.

Each project should contains below things
- **File which contains entry point function**
- **File which contains helper function**
- **File which works as header file**

1.Write a program which accept string from user and copy the contents of that string into another string. (Implement strcpy() function)

Input : **"Marvellous Multi OS"**

Output : **"Marvellous Multi OS" in another string**

```
void StrCpyX(char *src, char *dest)
{
	// Fileter
	while(*src != '\0')
	{
		*dest = *src;
		src++;
		dest++;
	}
}

int main()
{
	char arr[30] = "Marvellous Multi OS";
	char brr[30];		// Empty string
```

```
        StrCpyX(arr,brr);

        prnitf("%s",brr);           // Marvellous Multi OS
        return 0;
}
```

2. Write a program which accept string from user and copy the contents of that string into another string. (Implement strncpy() function)

Input : **"Marvellous Multi OS"**
10

Output : **"Marvellous**

Note : If third parameter is greater than the size of source string then copy whole string into destination.

```
void StrNCpyX(char *src, char *dest, int iCnt)
{
        // Fileter
        while((*src != '\0') && (icnt != 0) )
        {
                *_____ = *_____;
                _________ ++;
                _________ ++;
                _________  - -;
        }
}

int main()
```

```
{
        char arr[30] = "Marvellous Multi OS";
        char brr[30];           // Empty string

        StrNCpyX(arr,brr,10);

        prnitf("%s",brr);       // Marvellous

        return 0;
}
```

3. Write a program which accept string from user and copy capital characters of that string into another string.

Input : **"Marvellous Multi OS"**

Output : **"MMOS"**

```
void StrCpyCap(char *src, char *dest)
{
        // Fileter
        while(*src != '\0')
        {
                if((*src >= 'A') && (_________))
                {
                        *_______ = *_______;
                        _________ ++;
                }
                _________ ++;
        }
}
```

```
int main()
{
	char arr[30] = "Marvellous Multi OS";
	char brr[30];		// Empty string

	StrCpyCap(arr,brr);

	prnitf("%s",brr);	// MMOS

	return 0;
}
```

4. Write a program which accept string from user and copy small characters of that string into another string.

Input : **"Marvellous multi OS"**

Output : **"arvellous multi"**

```
void StrCpySmall(char *src, char *dest)
{
	// Fileter
	while(*src != '\0')
	{
		if((*src >= 'a') && (_______))
		{
			*_____ = *_____;
			_______ ++;
		}
		_______ ++;
	}
}
```

```
int main()
{
	char arr[30] = "Marvellous multi OS";
	char brr[30];		// Empty string

	StrCpySmall(arr,brr);

	prnitf("%s",brr);	// arvellous multi

	return 0;
}
```

5. Write a program which 2 strings from user and concat second string after first string. (Implement strcat() function).

Input : **"Marvellous Infosystems"**
"Logic Building"

Output : **"Marvellous Infosystems Logic Building"**

```
void StrCatX(char *src, char * dest)
{
	// Filter
	while(*str != _______)		// Traverse first string till end
	{
		// Logic
	}
	wihle(* dest != '/0')		// Copy contents of destination in source
	{
		// Logic
	}
```

```
        *dest = '/0';
}
int main()
{
        char arr[50] = "Marvellous Infosystems";
        char brr[30] = "Logic Building"

        StrCatX(arr,brr);

        prnitf("%s",arr);      // Marvellous Infosystems Logic Building

        return 0;
}
```

6. Write a program which 2 strings from user and concat N characters of second string after first string.Value of N should be accepted from user. (Implement strncat() function).

Note : If third parameter is greater than the size of second string then concat whole string after first string.

Input : **"Marvellous Infosystems"**
"Logic Building"
5

Output : **"Marvellous Infosystems Logic"**

```
void StrNCatX(char *src, char * dest, int iCnt)
{
        // Filter
        while(*str != _______)                  // Traverse first string till
end
        {
```

```
        // Logic
    }
    wihle((* dest != '/0') && (iCnt != 0))          // Copy contents of
destination in source
    {
        // Logic
        iCnt- -;
    }
    *dest = '/0';
}

int main()
{
    char arr[50] = "Marvellous Infosystems";
    char brr[30] = "Logic Building"

    StrNCatX(arr,brr,5);

    prnitf("%s",arr);                    // Marvellous Infosystems Logic
Building

    return 0;
}
```

7. Write a program which 2 strings from user and check whether contents of two strings are equal or not. (Implement strcmp() function).

Input : **"Marvellous Infosystems"**
"Marvellous Infosystems"
Output : **TRUE**

```
BOOL StrCmpX(char *src, char * dest)
{
	// Filter
	while((*str != _______) && (*dest != '/0'))
	{
		// Logic
	}

	if(________ || ________)
	{
		return TRUE;
	}
	else
	{
		return FALSE;
	}
}

int main()
{
	BOOL bret = TRUE;
	char arr[50] = "Marvellous Infosystems";
	char brr[30] = "Marvellous Infosystems"

	bret = StrCmpX(arr,brr);

	if(bret == TRUE)
	{
		prnitf("Both strings are equal");
	}
	else
	{
```

```
        prnitf("Both strings are not equal");
    }
    return 0;
}
```

8. Write a program which 2 strings from user and check whether first N contents of two strings are equal or not. (Implement strcmp() function).

Note : If third parameter is greater than the size of second string then concat whole string after first string.

Input : **"Marvellous Infosystems"**
"Marvellous Logic Building"
10

Output : **TRUE**

```
BOOL StrNCmpX(char *src, char * dest ,int iCnt)
{
    // Filter
    while(____________________)
    {
        // Logic
    }
    if(____________________)
    {
        return TRUE;
    }
    else
    {
        return FALSE;
    }
}
```

```
int main()
{
	BOOL bret = TRUE;
	char arr[50] = "Marvellous Infosystems";
	char brr[30] = "Marvellous Logic Building"

	bret = StrNCmpX(arr,brr,10);
	if(bret == TRUE)
	{
		prnitf("Both strings are equal");
	}
	else
	{
		prnitf("Both strings are not equal");
	}
	return 0;
}
```

9. Accept sing from user and reverse the contents of that string. (Implement strrev() function)

Input : "abcdef"

Output : "fedcba"

```
void StrRevX(char *str)
{
	// Filter
	char temp;
	char *first,*last;
	first = str;
	last = str;
	while(*last != '\0')
	{
```

```
            last++;
        }
        last- -;
        while(first <= last)
        {
            // Swapping logic
        }
}

int main()
{
        char arr[50] = "Marvellous";

        StrRevX(arr);

        prnitf("%s",arr);       // soullevraM

        return 0;
}
```

10. Accept string from user and check whether the string is palindrome or not without considering its case.

Input : **"abccBa"**

Output : **TRUE**

```
void StrPallindrome(char *str)
{
        // Filter
        // Logic
}
```

```
int main()
{
	BOOL bret;
	char arr[20] = "abccBa";
	bret = StrPallindrome(arr);
	if(bret == TRUE)
	{
		printf("String is pallindrome\n");
	}
	else
	{
		printf("String is not pallindrome\n");
	}
	return 0;
}
```