Logic Building Assignment : 12

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**

All below questions are depends on ASCII values of characters. Please consider below table to solve the questions.

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

1. Accept Character from user and check whether it is alphabet or not (A-Z a-z).

Input : F

Output : TRUE

Input : &

Output : FALSE

```
BOOL ChkAlpha(char ch)
{
      // Apply condition to check whether it is alphabet or not.
}
```

2. Accept Character from user and check whether it is capital or not (A-Z).

Input : F

Output : TRUE

Input : d

Output : FALSE

```
BOOL ChkCapital(char ch)
{
      // Apply condition to check whether it is capital or not.
}
```

3. Accept Character from user and check whether it is digit or not (0-9).

Input : 7

Output : TRUE

Input : d

Output : FALSE

```
BOOL ChkDigit(char ch)
{
      // Apply condition to check whether it is digit or not.
}
```

4. Accept Character from user and check whether it is small case or not (a-z).

Input : g

Output : TRUE

Input : D

Output : FALSE

```
BOOL ChkSmall(char ch)
{
      // Apply condition to check whether it is small case or not.
}
```

5. Accept division of student from user and depends on the division display exam timing. There are 4 divisions in school as A,B,C,D. Exam of division A at 7 AM, B at 8.30 AM, C at 9.20 AM and D at 10.30 AM. (Application should be case insensitive)

Input : C

Output : Your exam at 9.20 AM

Input : d

Output : Your exam at 10.30 AM

```
void DisplaySchedule(char chDiv)
{
      // Logic
}
```

6. Write a program which displays ASCII table. Table contains symbol, Decimal, Hexadecimal and Octal representation of every member from 0 to 255.

```
void DisplayASCII()
{
      // Logic
}
```

7. Accept two characters from user and swap its contents if both the characters are small or both the characters are capital. In other cases keep the contents as it is.

Input : K L

Output : L K

Input : K o

Output : K o

Input : u g

Output : g u

Input : * h

Output : * h

```
void SwapX(char *p, char *q)
{
      // Swap if condition is true
}
```

8. Accept character from user. If it is capital then display all the characters from the input characters till Z. If input character is small then print all the characters in reverse order till a. In other cases return directly.

Input : Q

Output : Q R S T U V W X Y Z

Input : m

Output : m l k j i h g f e d c b a

Input : 8

Output :

```
void Display(char ch)
{

}
```

9. Accept Character from user and check whether it is special symbol or not (!, @, #, $, %, ^, &, *).

Input : %

Output : TRUE

Input : d

Output : FALSE

```
BOOL ChkSpecial(char ch)
{
      // Apply condition to check whether it is special or not.
}
```

10. Accept character from user and display its ASCII value in decimal, octal and hexadecimal format.

Input : A

Output : Decimal 65
Octal 0101
Hexadecimal 0X41

```
void Display (char ch)
{
          // Logic
}
```