Part 1 – Loop Logic – 5 Points each

Q1 Output of Code Below:
Q1)__ 0 5 1 6 2 7 3 8

```
 int I = 0,J=5;
while (I*J <= 25)
{
        cout << I << " " << J << " ";
        I++;
        J++;
}
cout << endl;
```

Q2 Output of Code Below:
Q2)_ 1 2 1 3 2 1 _____

```
for (int i = 0; i <= 3; i++)
{
        int k = i;
        while (k)
        {
                cout << k << " ";
                k--;
        }
}
```

Q3 Output of Code Below:
Q3)_ 5 _____

```
int i = 5;
do
{
        while (i)
        {
                cout << i << " ";
                if (i > 1)
                        break;
        }
} while (i>5);
cout << endl;
```

Q4 Output of Code Below:

Q4) 0

```
for (int i = 0; i * i<100; i+=10)
{
        cout << i << " ";
}
cout << endl;
```

Q5 Output of Code Below:

Q5) 0 16 32 48 64

```
for (int i = 0; i < 10; i += 2)
{
        for (int j = 0; j < 10; j += 2)
        {
                if (i < 8)
                        break;
                cout << i * j << " ";
        }
}
```

Q6 Output of Code Below:

Q6) 11 12 6 7 12 13

```
int a = 10;
int f()
{
        a++;
        cout << a << " ";
        return(a);
}
int f1(int a=5)
{
        cout << a + 1 << " ";
        return(a+1);
}
int f2(int a)
{
        cout << f1() + 1 << " ";
        return(f() + 1);
}

int main()
{
        cout << f2(f1(f())) << endl;
        return(0);
}
```

Part 2 – C/C++ Knowledge – 5 Points Each

Q1 What does the following function do?
Q1___Returns true if a number is positive_____

```
bool f2(int i)
{
        return(i >= 0);
}
```

Q2 Is there any difference between the two loops below?
Q2 LOOP 1 will loop for any non zero value while LOOP 2 will only loop for positive values_____

LOOP 1

```
while(i)
  i--;
```

LOOP 2

```
while(i>0)
  i--;
```

Q3 What is the major difference between a while loop and a do while loop?
Q3__do while always executes at least one time_____

Q4 Usually when I print a string (text) go to the next line afterwards using "<< endl"
i.e. cout << "ABC" << endl;
Can I improve this code?
Q4) Create a helper function called PrintLine that takes a string as a parameter and always prints an endl_____

Part 3 – Programming – 10 points Each

```
Functions You can use with your code
int NumberOfDigits(int num)
{
	int NumOfDigits = 0;
	do // there is always at least one digit
	{
		NumOfDigits++;
		num /= 10;
	} while (num);
	return(NumOfDigits);
}

void PrintNumberRightAligned(int num, int width=-1)
{
	if(width==-1)
		width = NumberOfDigits(num) + 1;
	int numOfSpaces = width - NumberOfDigits(num);
	while (numOfSpaces>0)
	{
		cout << " ";
		numOfSpaces--;
	}
	cout << num;
}

void Swap(int &i,int &j)
{
	int temp = i;
	i = j;
	j = temp;
}
```

Q1) PrintCountingChart - Prints the first 100 numbers 1..100
10 numbers on a line the numbers are right justified.
See example output of first three rows

```
 1  2  3  4  5  6  7  8  9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
```

```
void PrintCountingChart()
{
	for (int row = 0; row < 100; row += 10)
	{
		for (int col = 1; col <= 10; col++)
		{
			PrintNumberRightAligned(row + col,4);
		}
		cout << endl;
	}

	// Method 2 from Mohaiminul Islam
	for (int num = 1; num <= 100; num++)
	{
		PrintNumberRightAligned(num,4);
		if(num%10==0)
			cout << endl;
	}

}
```

Q2 -Order - Orders three integers in ascending order
Note: You will probably need the Swap function above.
See example below.

```
int a = 7, b = 9, c = 8;
Order(a, b, c);
cout << a << b << c << endl; ==> 789
int a = 9, b = 7, c = 7;
Order(a, b, c);
cout << a << b << c << endl; ==> 779
void Order(int &a, int &b, int &c)
{
        // Method 1
        // is b the smallest?
        if((b <= a) && (b <= c))
                Swap(b,a);
        // is c the smallest?
        if ((c <= a) && (c <= b))
                Swap(c, a);

        // is c the next smallest
        if (c < b)
                Swap(c, b);

        // Method 2 (from Ethan and Julia)
        // Is b smaller than a
        if (a > b)
                Swap(a, b);
        // is c smaller than a (and b)
        if (a > c)
                Swap(a, c);
        // is c smaller than b
        if (b > c)
                Swap(b, c);

}
```

Q3) EveryPossibleFourCharacters Prints every possible combination of the four letters where each letter is between A..Z.
Output of first 10 combinations:

```
AAAA
AAAB
AAAC
AAAD
AAAE
AAAF
AAAG
AAAH
AAAI
```

AAAJ

```
void EveryPossibleFourCharacters()
{
        for (char a = 'A'; a <= 'Z'; a++)
                for (char b = 'A'; b <= 'Z'; b++)
                        for (char c = 'A'; c <= 'Z'; c++)
                                for (char d = 'A'; d <= 'Z'; d++)
                                        cout << a << b << c << d << endl;
}
```

Q4) Translate the following for loop into a while loop

Which has the exact same logic

```
for (int i = 10, j = 0; i > 5 && j < 10; i++, j++)
{
        if (j == 5)
                continue;
        cout << i << " " << j << endl;
}
```

Q4 ANSWER

Q5) RandomGrade - Returns a Random Grade and sign
First three lines have been written for you.
Sample Output

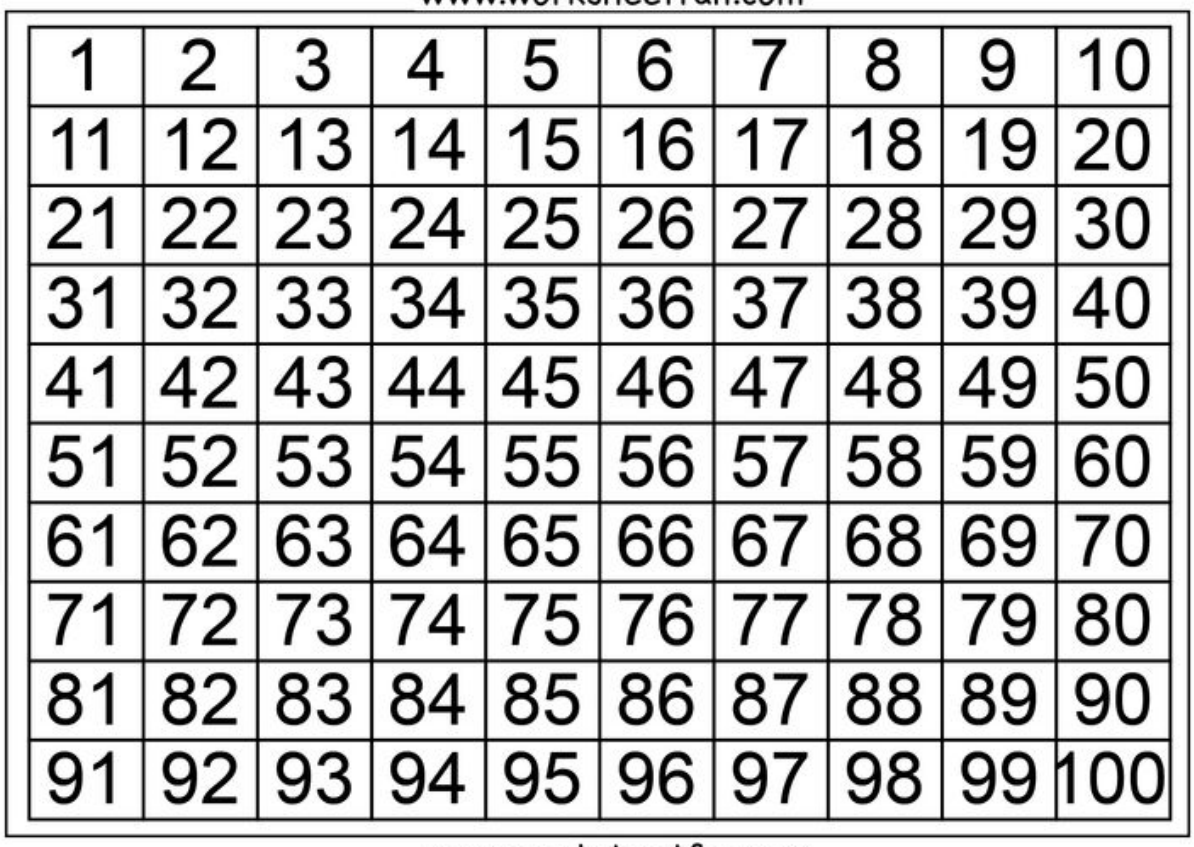
```
RandomGrade(grade, sign) grade ==> 'A' sign = '-'
RandomGrade(grade, sign) grade ==> 'F' sign = ' '
RandomGrade(grade, sign) grade ==> 'C' sign = '+'
void RandomGrade(char &grade,char &sign)
{
        srand(time(0));
        char grades[6] = "ABCDF";
        char signs[4] = "+- ";
        grade = grades[rand() % 5];
        sign = signs[rand() % 3];
}
```

ASCII CHART

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040	&#032;	Space	64	40	100	&#064;	@	96	60	140	&#096;	`
1	1	001	Start of Header	33	21	041	&#033;	!	65	41	101	&#065;	A	97	61	141	&#097;	a
2	2	002	Start of Text	34	22	042	&#034;	"	66	42	102	&#066;	B	98	62	142	&#098;	b
3	3	003	End of Text	35	23	043	&#035;	#	67	43	103	&#067;	C	99	63	143	&#099;	c
4	4	004	End of Transmission	36	24	044	&#036;	$	68	44	104	&#068;	D	100	64	144	&#100;	d
5	5	005	Enquiry	37	25	045	&#037;	%	69	45	105	&#069;	E	101	65	145	&#101;	e
6	6	006	Acknowledgment	38	26	046	&#038;	&	70	46	106	&#070;	F	102	66	146	&#102;	f
7	7	007	Bell	39	27	047	&#039;	'	71	47	107	&#071;	G	103	67	147	&#103;	g
8	8	010	Backspace	40	28	050	&#040;	(	72	48	110	&#072;	H	104	68	150	&#104;	h
9	9	011	Horizontal Tab	41	29	051	&#041;	)	73	49	111	&#073;	I	105	69	151	&#105;	i
10	A	012	Line feed	42	2A	052	&#042;	*	74	4A	112	&#074;	J	106	6A	152	&#106;	j
11	B	013	Vertical Tab	43	2B	053	&#043;	+	75	4B	113	&#075;	K	107	6B	153	&#107;	k
12	C	014	Form feed	44	2C	054	&#044;	,	76	4C	114	&#076;	L	108	6C	154	&#108;	l
13	D	015	Carriage return	45	2D	055	&#045;	-	77	4D	115	&#077;	M	109	6D	155	&#109;	m
14	E	016	Shift Out	46	2E	056	&#046;	.	78	4E	116	&#078;	N	110	6E	156	&#110;	n
15	F	017	Shift In	47	2F	057	&#047;	/	79	4F	117	&#079;	O	111	6F	157	&#111;	o
16	10	020	Data Link Escape	48	30	060	&#048;	0	80	50	120	&#080;	P	112	70	160	&#112;	p
17	11	021	Device Control 1	49	31	061	&#049;	1	81	51	121	&#081;	Q	113	71	161	&#113;	q
18	12	022	Device Control 2	50	32	062	&#050;	2	82	52	122	&#082;	R	114	72	162	&#114;	r
19	13	023	Device Control 3	51	33	063	&#051;	3	83	53	123	&#083;	S	115	73	163	&#115;	s
20	14	024	Device Control 4	52	34	064	&#052;	4	84	54	124	&#084;	T	116	74	164	&#116;	t
21	15	025	Negative Ack.	53	35	065	&#053;	5	85	55	125	&#085;	U	117	75	165	&#117;	u
22	16	026	Synchronous idle	54	36	066	&#054;	6	86	56	126	&#086;	V	118	76	166	&#118;	v
23	17	027	End of Trans. Block	55	37	067	&#055;	7	87	57	127	&#087;	W	119	77	167	&#119;	w
24	18	030	Cancel	56	38	070	&#056;	8	88	58	130	&#088;	X	120	78	170	&#120;	x
25	19	031	End of Medium	57	39	071	&#057;	9	89	59	131	&#089;	Y	121	79	171	&#121;	y
26	1A	032	Substitute	58	3A	072	&#058;	:	90	5A	132	&#090;	Z	122	7A	172	&#122;	z
27	1B	033	Escape	59	3B	073	&#059;	;	91	5B	133	&#091;	[	123	7B	173	&#123;	{
28	1C	034	File Separator	60	3C	074	&#060;	<	92	5C	134	&#092;	\	124	7C	174	&#124;	|
29	1D	035	Group Separator	61	3D	075	&#061;	=	93	5D	135	&#093;	]	125	7D	175	&#125;	}
30	1E	036	Record Separator	62	3E	076	&#062;	>	94	5E	136	&#094;	^	126	7E	176	&#126;	~
31	1F	037	Unit Separator	63	3F	077	&#063;	?	95	5F	137	&#095;	_	127	7F	177	&#127;	Del

asciicharstable.com

www.worksheetfun.com



1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

www.worksheetfun.com