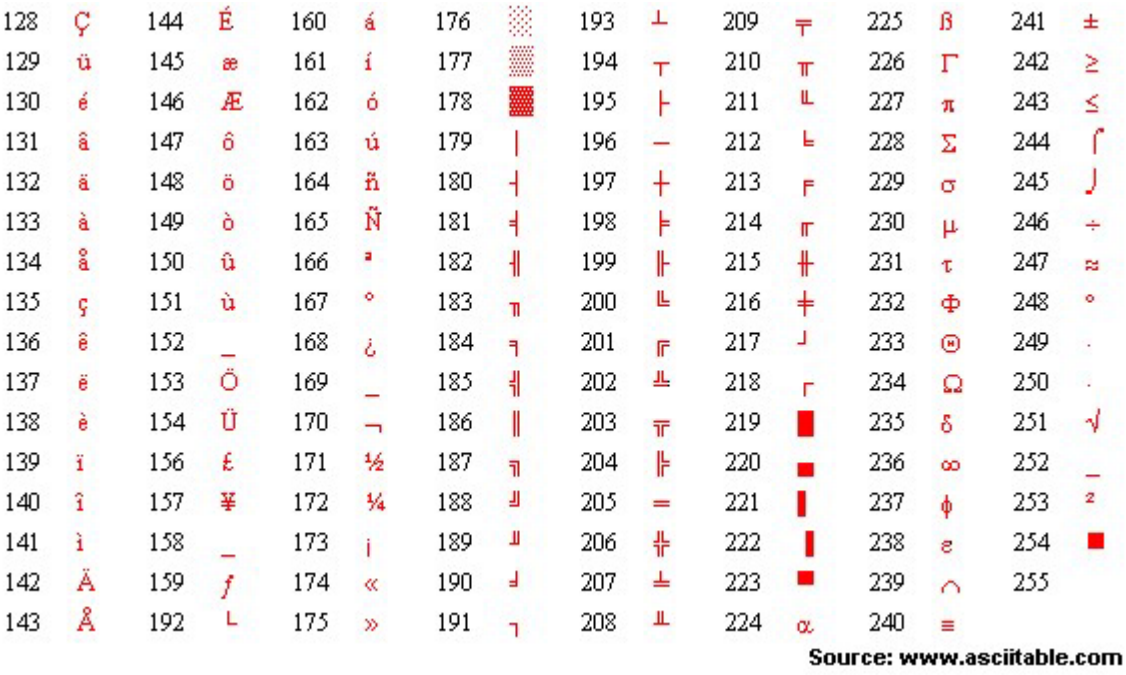
CMPS 150 Workbook: Chapter 2

ASCII Character Set Values

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.asciitable.com

Extended ASCII Character Set Values



128	Ç	144	É	160	á	176	▒	193	┴	209	╤	225	ß	241	±
129	ü	145	æ	161	í	177	▒	194	┬	210	╥	226	Γ	242	≥
130	é	146	Æ	162	ó	178	▓	195	├	211	╙	227	π	243	≤
131	â	147	ô	163	ú	179	│	196	─	212	╘	228	Σ	244	⌠
132	ä	148	ö	164	ñ	180	┤	197	┼	213	╒	229	σ	245	⌡
133	à	149	ò	165	Ñ	181	╡	198	╞	214	╓	230	µ	246	÷
134	å	150	û	166	ª	182	╢	199	╟	215	╫	231	τ	247	≈
135	ç	151	ù	167	º	183	╖	200	╚	216	╪	232	Φ	248	°
136	ê	152	_	168	¿	184	╕	201	╔	217	┘	233	⊕	249	·
137	ë	153	Ö	169	_	185	╣	202	╩	218	┌	234	Ω	250	·
138	è	154	Ü	170	¬	186	║	203	╦	219	▉	235	δ	251	√
139	ï	156	£	171	½	187	╗	204	╠	220	▄	236	∞	252	_
140	î	157	¥	172	¼	188	╝	205	═	221	▌	237	ϕ	253	²
141	ì	158	_	173	¡	189	╜	206	╬	222	▐	238	ε	254	■
142	Ä	159	ƒ	174	«	190	╛	207	╧	223	▀	239	∩	255	
143	Å	192	└	175	»	191	┐	208	╨	224	α	240	≡		

Source: www.asciitable.com

Section 2.1: Boolean Expressions

1) Determine if the following Boolean expressions are true or false:

a) (true && true)
b) (false && true)
c) (true || true)
d) (false || true)
e) (!true)
f) (!(!false))
g) (3 < 7)
h) (3 + 4 <= 6 - 1)
i) (3.4 != 3.3)
j) (3 > 2 && 5 < 7)
k) (5 >= 9 || 15 < 20)
l) (4 > 3 && 5 > 2 && 5 > 6)
m) (9 < 8 || 8 < 7 || 6 < 7)
n) ((1 < 2 && 3 < 2) || (13 < 14))
o) ('a' == 'b')
p) ("Fred" != "Sam")

2) Determine if the following Boolean expressions are true or false:

a) (true && false)
b) (false && false)
c) (true || false)
d) (false || false)
e) (!false)
f) (!(!true))
g) (4 > 5)
h) (5 - 2 >= 3 * 2)
i) (7.9 == 8.0)
j) (6 <= 7 && 6 >= 8)
k) (17 < 2 || 17 > 22)
l) (1 < 2 && 2 < 3 && 3 < 4)
m) (45 == 4 || 13 < -5 || 33 != 7)
n) ((1 !=1) || (87 > 1 && 99 > -15))
o) ('8' != '+')
p) ("Wanda" == "Joy")

3) Given that *max* is 20 and *min* is 10, determine if the following Boolean expressions are true or false:

a) (max >= min)
b) (max >= 14 && max <= 34)
c) ((min > 10 || min == 10) == (min >= 10))

4) Given that *max* is 20 and *min* is 10, determine if the following Boolean expressions are true or false:

a) (max != min)
b) (min < 23 && min > 7)
c) ((max < 17 || max > 3) == (max > min))

5) Given that *password* has the value "secret" and *guess* has the value "secret", determine if the following Boolean expressions are true or false.

a) (password == "let me in")
b) ("SECRET" != password)
c) (secret == guess)

6) Given that *firstName* has the value " François" and *lastName* has the value "Begnaud", determine if the following Boolean expressions are true or false.

a) (firstName == lastName)
b) (firstName != "Gerard")
c) (lastName == "Begnaud" && firstName == "François")

Section 2.2: Branching Mechanisms

1) Given that *a* is 1, *b* is 2 and *c* is 3, what is the output of the following code?

a)
```
cout << "The Beginning!" << endl;
if (a > 2)
    cout << "A One" << endl;
cout << "The End!" << endl;
```

b)
```
d = 99;
if (a < 5)
{
     cout << "a is less than 5" << endl;
     d = 14;
}
cout << "d is " << d << endl;
```

c)
```
if (a > c - b)
     z = 17;
else
     z = -17;
cout << "z is " << z << endl;
```

d)
```
if (12 / c == 4)
  {
     j = 3;
     k = 5;
}
else
{
     j = 10;
     k = 15;
}
cout << j + k << endl;
```

e)
```
if (a > b)
   cout << "a is greater than b" << endl;
else if (a == b)
     cout << "a is equal to b" << endl;
else
     cout << "a is less than b" << endl;
```

f)
```
(a < b) ? m = 45 : m = 55;
cout << m;
```

g)
```
case(a + b)
   {
       case 3 :
           cout << "The ";
       case 2 :
           cout << "power ";
           break;
       case 4 :
           cout << "of ";
           break;
       default:
           cout << "cheese!";
   }
```

2) Given that *a* is 1, *b* is 2 and *c* is 3 , what is the output of the following code?

a)
```
cout << "spring" << endl;
if (b <= 2)
    cout << "summer" << endl;
cout << "fall" << endl;
```

b)
```
e = 10;
if (a*b < c)
{
    e = 20;
    cout << "a times b is less than c" << endl;
}
cout << "e is " << e << endl;
```

c)
```
if (a != b && b > -5)
    h = a + b + c;
else
    h = -a - b - c;
cout << "h is " << h << endl;
```

d)
```
if (b * c != 6)
{
    width = 9;
    length = 3;
}
else
{
    width = 4;
    length = 5;
}
cout << "Area: " << width * length << endl;
```

e)
```
if (c < b + a)
      j = 100;
   else if (c == b + a)
      j = 200;
   else
     j = 300;
```

f)
```
(b > c) ? n = a + b : n = a + c;
cout << n;
```

g)
```
case (c - 1)
{
    case 3 :
        cout << "vini ";
        break;
    case 2 :
        cout << "vidi ";
    case 1 :
        cout << "vici ";
        break;
    default:
        cout << "dixi";
}
```

3) Write an *if* statement to change any negative number stored in *num* to positive.

4) Write an *if* statement to output the hours of overtime (time worked over forty hours in a week) worked if *hours* has a value greater than 40.

5) Write an *if-else* statement that will output "hot" if the value in *temp* is over 85 and will output "not hot" if the value in *temp* is 85 or below.

6) Write an *if-else* statement that will output "school zone" if *speedLimit* has a value of 30 or less, but otherwise outputs "drive like heck".

7) Write a *nested if-else* statement that will output "hot" if the value in *temp* is above 85, "OK" if the value in temp is at least 45 but less than or equal to 85, and "cold" if the value in *temp* is less than 45.

8) Write a *nested if-else* statement that will output "school zone" if *speedLimit* has a value of 30 or less, "watch for the cop" if *speedLimit* has a value of above 30, but less than or equal to 75, and "drive like heck" if the value in *speedLimit* is greater than 75.

9) Write a *switch* statement that outputs "red" if the value of *color* is 'r', "green" if *color* has the value 'g', "yellow" if *color* has the value 'y', and "blue" if *color* has the value 'b'. If none of the above applies, the output should be "white".

10) Write a *switch* statement that assigns the value of 1 to *qualityPoints* if the value in *grade* is 'D', assigns the value of 2 to *qualityPoints* if the value in *grade* is 'C', assigns the value of 3 to *qualityPoints* if the value in *grade* is 'B', and assigns the value of 4 to *qualityPoints* if the value in *grade* is 'A'. If none of the above applies, *qualityPoints* should be assigned the value of 0.

11) Write a program that outputs "Go on in!" if the password the user enters is the same as the password stored in the program. (Make up a password for your program.)

12) Write a program that outputs "the bag is overweight" if the user enters a bag weight of more than 40 pounds.

13) Write a program that outputs "Go on in!" if the user enters a correct password or "You May Not Enter!" if the users password does not match the programs password. (Make up a password for your program.)

14) Modify program #12 so that it outputs "the bag is overweight" if the user enters a bag weight of more than 40 pounds, otherwise it outputs "the bag weight is within limits".

15) Hourly workers at McFudd’s are paid by the hour. If a worker works more than 40 hours in a week, the hourly pay rate for the hours past the first 40 hours are increased by 50%. This is known as “time and a half” overtime pay. For example, if a worker making $6.00 per hour works 45 hours in one week, that worker will receive 45.0 * 6.0 + 6.0 * 5.0 * .5, or $285.00 in pay. Write a program that asks for and receives the hours worked and rate of pay from the user. The program should output the correctly calculated amount of pay.

16) General Forge and Foundry pays its sales people by commission. If a salesperson sells less than $10,000 in a week the commission is 4% of the sales. Otherwise, if the salesperson sells at least $10,000 in a week, the commission is 5%. Write a program that will accept the amount of sales in a week from the user, then outputs the amount of commission earned by the salesperson.

17) Create a program that receives a value *n* from the user and returns the value calculated by (2*n*n) if the value of *n* is less than 0, 1 if the value of *n* is 0, and (-2*n*n) if the value of *n* is greater than 0.

18) Given a value of the variable *year*, the date for Easter can be computed as follows:

```
a = year % 19;
b = year % 4;
c = year % 7;
d = (19 * a + 24) % 30;
e = (2 * b + 4 * c + 6 * d + 5) % 7;
date = 22 + d + e;
```

If these calculations are performed, *date* will contain the date of Easter for the year. However, this formula returns a value based on the number of days in March. This is fine if the value of *date* is less than 32. For example, if *date* had the value 27, this would indicate that Easter would fall on March 27 of the year given. But, if *date* contains the value 36, this would indicate that Easter would fall on March 36 of the year give, which is impossible. Obviously, March 36 is really April 5.

Write a program that accepts a year from the user, then computes and outputs the date of Easter with the correct month, in the form:

month day, year

19) A year is a leap year if it meets the following criteria:

- the year is greater than 1582
- the year is evenly divisible by 4
- if the year is evenly divisible by 100, it must also be evenly divisible by 400

Write a program that receives the year in question from the user and outputs a message stating if the year is a leap year or not a leap year.

20) Create a program that accepts a letter from the user and returns the equivalent digit on the dial or keypad of a telephone. Telephone letters match digits as follows:

```
ABC = 2 DEF = 3 GHI = 4 JKL = 5
MNO = 6 PRS = 7 TUV = 8 WXY = 9
```

Notice that the letter Q is not used and does not need to be accounted for (can be ignored). The function should work with upper and lower case letters.

21) Sales personnel at the J&R Widget Company are paid a commission each week based on the amount of gross sales that each sales person is responsible for. If a sales person's gross sales range from $0 to $10,000, the commission rate is 2.5%. If the sales person's gross sales are more than $10,000 but less than $20,000, the commission rate is 3.5%. If the sales person's gross sales are at least $20,000, the commission rate is 4.7%. In addition to the commission, any sales person selling more than $40,000 in one week is awarded a $200 bonus and any sales person selling more than $50,000 in one week is awarded an additional $300 bonus.

Write a program that prompts for and accepts the gross sales for one week of a sales person and outputs how much he/she should receive in commissions and bonuses

22) The Florida election commission is having difficulty determining which of three candidates won an election for dogcatcher or if there should be a run off election and, if so, who should be in it. (Since there are no Democrats in the race, the commission is having trouble because there is no obvious group of ballots to exclude.)

Write a program that prompts and accepts the names of the three candidates and the amount of votes each received. If one candidate received more than 50% of the vote, declare that candidate the winner. If no one won outright, output the two candidates with the highest number of votes (i.e. the candidates that will face each other in the runoff).

Section 2.3: Loops - while and do-while

1) What is the output of the following code?

a)
```
int count = 0;
count++;
count++;
cout << count << endl;
```

b)
```
double total = 0.0;
total = total + 15;
total = total * 2.0;
total = total / 4;
cout << total << endl;
```

c)
```
int wishes = 10;
wishes += 4;
wishes /= 2;
wishes *= 3;
cout << wishes << endl;
```

2) What is the output of the following code?

a)
```
int numThings = 10;
count--;
count--;
cout << count << endl;
```

b)
```
double sum = 20.0;
sum = sum - 5;
sum = sum / 3;
sum = sum * 5;
cout << sum << endl;
```

c)
```
int numbers = 55;
numbers /= 5;
numbers += 1;
numbers *= 2;
cout << numbers << endl;
```

3) What is the output of the following code?

a)
```
int a = 10;
while (a > 0)
{
    a--;
    cout << a << ' ';
}
```

b)
```
int z = 5;
while (z < 20)
{
    cout << z << ' ';
    z += 4;
}
```

c)
```
int b = 2;
do
{
    b = b * 2;
    cout << b << endl;
}
while (b < 15);
cout << b << endl;
```

d)
```
int sum = 0, num = 0;
while (num < 5)
{
    num++;
    sum += num;
}
cout << sum << endl;
```

e)
```
int a = 1, b;
while (a < 5)
{
    b = 1;
    while (b < 3)
    {
        cout << a << b << endl;
        b++;
    }
    a++;
}
```

f)
```
int g = 5, h;
while (g > 3)
{
    h = 9;
    g--;
    while (h > 7)
    {
        h--;
        cout << g << h << endl;
    }
}
```

4) What is the output of the following code?

a)
```
int x = 0;
while (x < 5)
{
    cout << x << ' ';
    x++;
}
```

b)
```
int y = 8;
while (y < 20)
{
    y += 3;
    cout << y << ' ';
}
```

c)
```
int m = 10;
do
{
    cout << m << ' ';
    m = m * 3;
}
while (m < 100)
cout << m << endl;
```

d)
```
double total = 0.0, count = 3.0;
while (count > 0)
{
    total = total + count;
    count = count - .5;
}
cout << total << endl;
```

e)
```
int x = 5, y;
while (x < 8)
{
    y = 11;
    while (y < 13)
    {
        cout << x << y << endl;
        y++;
    }
    x++;
}
```

f)
```
int m = 15, n;
while (m > 11)
{
    n = 19;
    m--;
    while (n > 17)
    {
        n--;
        cout << m << n << endl;
    }
}
```

5) Answer the following:

a) If the user tries to enter the values: 1 2 3 4 5 6 -2
What is output?

```
int a = 0, b;
do
{
    cout << “enter a number: “;
    cin >> b;
    a = a + b;
}
while (b > 0);
cout << a;
```

b) If the user tries to enter the values: 1 2 3 4 5 6 -2
What is output?

```
int a = 0, b;
cout << “enter a number: “;
cin >> b;
while (b > 0)
{
    a = a + b;
    cout << “enter a number: “;
    cin >> b;
}
cout << a;
```

c) What characters would cause this loop to halt?

```
char c;
do
{
    cout << “Elevator, Inc. Enter Floor Number: “;
    cin >> c;
}
while (c != ‘1’  &&  c != 2  &&  c != ’3’);
```

6) Answer the following:

a) If the user tries to enter the values: 1 2 3 4 5 6 -2
What is output?

```
int a = x, y;
do
{
    cout << “enter a number: “;
    cin >> y;
    x = x + y;
}
while (y < 6);
cout << x;
```

b) If the user tries to enter the values: 1 2 3 4 5 6 -2
What is output?

```
int x = 0, y;
cout << “enter a number: “;
cin >> y;
while (y < 6)
{
    x = x + b;
    cout << “enter a number: “;
    cin >> y;
}
cout << x;
```

c) What characters would cause this loop to halt?

```
char pick;
do
{
    cout << “Food-o-mat a-green milk  b-mystery meat” << endl
         << “           c-gelatin     d-brown lettuce” << endl;
    cin >> pick;
}
while (pick != ‘a’  &&  pick != ‘b’  &&  pick != ’c’  &&
       pick != ‘d’);
```

7) Write a program that uses a while loop to output the odd integers from -99 to 99.

8) Write a program that uses a while loop to output the even integers from -100 to 100.

9) Write a program that accepts the heights in inches of 5 people from the user, then outputs the average height.

10) Write a program that accepts the weight in pounds of 5 packages, then outputs the average weight.

11) Write a program that accepts the heights in inches of an unknown number of people (i.e. the user will determine when the program has input enough people), then outputs the average height. The program should run until the user inputs a height less than 0.

12) Write a program that accepts the weight in pounds of an unknown number of packages (i.e. the user will determine when the program has input enough weights of packages), then outputs the average weight. The program should run until the user inputs a weight less than 0.

Section 2.3: Loops - for

1) What is the output of the following code?

```
for (int x=0; x<10; x++)
    cout << x * x << ' ';
```

2) What is the output of the following code?

```
for (int y=5; y>0; y--)
    cout << y * 3 << ' ';
```

3) What is the output of the following code?

```
for (int a=1; a<5; a++)
{
    cout << a << ' ';
    for (int b=1 b<3; b++)
        cout << b << ' ';
    cout << endl;
}
```

4) What is the output of the following code?

```
for (int h=21; h>19; h--)
{
    for (int g=4; g<6; g++)
        cout << g << ' '
    cout << h << endl;
}
```

5) What is the output of the following code?

```
for (int a=0; a<2; a++)
{
    for (int b=0; b<2; b++)
         for (int c=0; c<2; c++)
             cout << a << b << c;
    cout << endl;
}
```

6) What is the output of the following code?

```
for (int w=0; w<2; w++)
    for (int x=0; x<2; x++)
        for (int y=0; y<2; y++)
            for (int z=0; z<2; z++)
                cout << z;
```

7) What is the output of the following code?

```
int m=6, sum=0;
for (; m<10; m++)
    sum += m;
cout << "m is " << m << endl;
cout << "sum is " << sum << endl;
```

8) What is the output of the following code?

```
double d=0.0;
for (; d<3.0; d+=.5)
    cout << d << ' ';
cout << d << endl;
```

9) What is the output of the following code?

```
for (char c='A'; c<='Z'; c++)
    cout << c;
cout << endl;
```

10) What is the output of the following code?

```
for (char n='0'; n<='9'; n++)
    cout << n;
cout << endl;
```

11) Write a program that uses a *for loop* to output your name 200 times.

12) Write a program that uses a *for loop* to output every integer from –100 to 100.

13) Write a program that uses a *for loop* to compute and output the cubes of the integers from –3 to 3.

14) Write a program that uses a *for loop* to output all possible solutions to the expression $X^2 - y^2$ where *X* is an integer from 3 to 9 and *Y* is an integer from 2 to 4.

15) Write a program that uses *for loops* to output all possible solutions for the expression $6X^3 - 5Y^2 - 4Z + 2$ where *X* is an integer from 1 to 20, *Y* is an integer from –5 to 5, and *Z* is an integer from 3 to 17.

16) Write a program that receives a positive integer *n* from the user and outputs the sum of the integers from 1 to *n*. The program should use a *for loop.*

17) Write a program that receives a positive integer n from the user and outputs the factorial of the *n*. (factorial of *n*, written n!, is 1 * 2 * 3 * 4 * … * n) The program should use a *for loop.*

18) Write a program that receives two integer numbers *n* and *m*, then returns the sum of the integers from *n* to *m.* The program should use a *for loop.*

19) Write a program that receives an integer *n* from the user and a positive integer power *p* from the user. The program will compute *n* raised to the power *p* with a counter and *for-loop*, then outputs the value of *n* raised to the power *p*.

20) Write a program that sums the heights in inches of any number of people (i.e. the number of people will be determined by the program user), then outputs the average height. (Hint: The Boolean expression in the for loop can test for an exit value.)

21) Write a program that receives a prime number *P* by the user, outputs its Mersenne number. (A Mersenne number is defined as $2^P - 1$ where *P* is a prime number.) The program should use a *for loop.*

22) In C++, characters are actually a special case of integer. Each character in the type *char* is a representation of a numeric value from 0 to 255. If a *char* variable is assigned an integer value, the equivalent character will be created in the *char* variable. For example, if 65 is assigned to the char variable c, when c is output, the character A will appear.

```
int i = 65;
char c = i;
cout << c << " is " << i;
```

Output: A is 65

Write a program that produces all 256 characters along side the numeric value of each character. The program should use a *for loop.*

23) Write a program that uses text characters to output an old style graphics text box with your name in it. The program should use a *for loop.*

Example:

24) Write a program to simulate 1000 roles of a pair of dice. Keep count of the number of times the result of the role results in a two, three, four, etc. When the simulated roles are completed, the program should output the number of times that a two was rolled, the number of times a three was rolled, etc.

Note: How to Generate Random Die Roles

a) include the standard header files <ctime> and <cstdlib>

b) before the loop, seed the random number algorithm

```
srand( (unsigned)time(NULL) );
```

c) use the rand( ) functions and the remainder operator (%) to generate numbers from 1 to 6 for each die

```
die1 = rand( ) % 6 + 1

    // rand( ) generates integers from 0 to 32,767. "% 6" is
    // used to limit the numbers generated to 0 to 5. Adding
    // one moves the number range to 1 to 6.
```

d) add the amount given by both dies together to get the amount of the role

The program should use a *for loop.*

25) Fibonacci numbers are the number series

0, 1, 1, 2, 3, 5, 8, 13, 21, …

The function Fibonacci is defined as:

```
f(0) = 0
f(1) = 1
f(n) = f(n-1) + f(n-2), where n is a integer greater than 1
```

In English, this means that:

```
giving 0 to the Fibonacci function returns 0,
giving 1 to the Fibonacci function returns 1,
giving 2 to the Fibonacci function returns 0 + 1 = 1,
giving 3 to the Fibonacci function returns 1 + 1 = 2,
giving 4 to the Fibonacci function returns 1 + 2 = 3,
giving 5 to the Fibonacci function returns 2 + 3 = 5,
etc.
```

In other words, each Fibonacci number *n* (where *n* is an integer greater than 1) is the sum of the two previous Fibonacci numbers.

Write a program that receives an integer *n* from the user and returns the *nth* Fibonacci number.

26) The number *PI* can be approximated in decimal numbers by the following function.

```
f(1) = 4 * (1 / (2*3*4)) + 3
f(2) = 4 * (1 / (2*3*4 – 1 / (4*5*6) ) + 3
f(3) = 4 * (1 / (2*3*4 – 1 / (4*5*6) + 1 / (6*7*8)) + 3
f(4) = 4 * (1 / (2*3*4 – 1 / (4*5*6) + 1 / (6*7*8) – 1 / (8*9*10)) + 3
f(5) = 4 * (1 / (2*3*4 – 1 / (4*5*6) + 1 / (6*7*8) – 1 / (8*9*10) + 1 /(10*11*12))
       + 3
f(6) = 4 * (1 / (2*3*4 – 1 / (4*5*6) + 1 / (6*7*8) – 1 / (8*9*10) + 1 / (10*11*12)
       – 1 / (12*13*14)) + 3
etc.
```

Write a program that receives an integer *n* (which stands for the number of times 1 is divided by increasingly larger numbers) and returns an approximation of *PI* based on that *n*. *(Hint: Be sure and specify the number of decimals so to output the best possible answer.)*