

# Checking for Various Types of Input Tutorial

## Sample Programs List

Sample Program: Search specific range of characters

Sample Converting Char to Int

Sample Program: Check for Non-Integer Values (only lets user enter Integer Values)

## ASCII Keyboard Codes

Definition: (American Standard Code for Information Interchange) is the most common format for text files in computers and on the Internet. In an ASCII file, each alphabetic, numeric, or special character is represented with a 7-bit binary number (a string of seven 0s or 1s). 128 possible characters are defined.

It is possible for a programmer to limit the type of inputs that comes from a keyboard. Picture below shows the keystroke to ASCII Code conversion. Using an if statement can limit what good/bad input could be based upon the input from the keyboard.

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

### Sample Program: Search specific range of characters

**//program works only if one character is pressed at a time (I.E 0, 1,2, a, b, 3, etc.) will not work with numbers over 10 or strings**

```
#include <iostream>
using namespace std;
int main()
{
char exit, error;
char myarray[5];           // creates an array that will hold characters
int counter=0;
for (int x=0; x <= 4; x++)
{
cout << "Enter a letter: " << " , ";
cin >> myarray[x];
if (myarray[x] >= 48 && myarray[x] <=57) // Eliminates inputs in the 49 to 57 area of the ASCII Code (Numbers 0-9)
{
// If a key is pressed (0-9) is pressed in this range the user will be prompted to
do // type in a new letter
{
cout << "Enter a new letter: ";
cin >> myarray[x];
counter++;
if (counter == 3) // Only gives the user 3 tries to input a proper letter
{
error = 'x'; // variable error to be used to compare in the do/while condition statement
x=5; // Sets counter in the FOR Loop to 5. Exiting out of the loop
}
}
while ( (myarray[x] >= 48 && myarray[x] <=57) && error != 'x');
}
}
for (int x=0; x <= 4; x++)
{
cout << myarray[x] << " , ";
}
cin >> exit;
return 0;
}
```

## Sample Converting Char to Int

//program works only if one character is pressed at a time (I.E 0, 1,2, a, b, 3, etc.) will not work with numbers over 10 or strings

```
#include <sstream>
```

```
# include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    stringstream str;
```

```
    char exit;
```

```
    string ans;
```

```
    cout << "enter a number: ";
```

```
    cin >> ans;
```

```
    str << ans;
```

```
    double x;
```

```
    str >> x;
```

```
    cout << x;
```

```
    cin >> exit;
```

```
return 0;
```

```
}
```

## Sample Program: Check for Non-Integer Values

```
#include<iostream>
using namespace std;

bool cond;

int main()
{
    char exit;
    int n;

    do
    {
        cout << "Enter an integer number:";
        cin >> n;

        cond = cin.fail(); // checks to see if the value is integer with variable bool cond. If
                           //false then it will reset the entry with the cin.clear(). True
                           //skips over the clear.

        cin.clear(); //clears the inputted state back to before a value was entered
        cin.ignore(INT_MAX, '\n');

    }while(cond); //loop will run until a good entry is typed in.

    cin >> char;
    return 0;
}
```