**Module flush()**

Set hasextra = 0

DOWHILE Set s = Call Module fgetc(stdin) != ‘\n’ AND s != EOF

Set hasextra = 1

ENDDO

Return hasextra

**END**

**Module ichecker(max, min, allowzero, psfix)**

Set error = 1

DOWHILE error == 1

Set errno = hasdecimal = haschar = isempty = isoverflow = hasspace = 0

IF psfix == 0 THEN

Display \_string, min, max

ELSE

IF psfix == 1

Display \_string, \_units, min, max

ELSE

IF psfix == 2

Display \_string, min, max, \_units

ELSE

Display \_string, min, max, SPACE, \_units

ENDIF

ENDIF

ENDIF

Call Module fgets(input, 40, stdin)

Set pChar = Call Module strchr(input,’\n’)

IF pChar != 0 THEN

Set length = Call Module strlen(input) – 1

IF input[0] == ‘\n’ THEN

Set isempty = 1

ENDIF

ELSE

IF Call Module fgetc(stdin) == ‘\n’ THEN

Set length = Call Module strlen(input)

ELSE

Set isoverflow = 1

Call Module flush()

ENDIF

ENDIF

IF isoverflow != 1 THEN

DO i = 0 TO i < length

IF Call Module isdigit(input[i]) == 0 THEN

IF i != 0 AND input[i] == ‘.’ AND hasdecimal == 0

Set hasdecimal = 1

continue

ELSE

IF i == 0 AND (input[i] == ‘+’ OR input[i] == ‘-‘) THEN

continue

ELSE

IF input[i] == SPACE OR input[i] == ‘\t’ THEN

Set hasspace = 1

ENDIF

ENDIF

ENDIF

Set haschar = 1

ENDIF

ENDDO

ENDIF

IF isempty == 1 THEN

Display “Detected input is empty, did you key in anything?”

ELSE

IF hasspace == 1 THEN

Display “Space character detected, please try again.”

ELSE

IF isoverflow == 1 THEN

Display “Input exceeds buffer length, please try again.”

ELSE

IF haschar == 0 THEN

Set value = Call Module strtol(input, NULL,10)

IF value == 0 AND allowzero == 0 THEN

Display “Value detected is 0 and invalid. Please try again.”

ELSE

IF errno == ERANGE OR value < min OR value > max THEN

Display “Value keyed in is out of range, should be between ‘min’ and ‘max’, please try again.”

ELSE

Set error = 0

ENDIF

ENDIF

ELSE

Display “Character detected, try inputting only numerical values.”

ENDIF

ENDIF

ENDIF

ENDIF

ENDDO

Return value

**END**

**Module fchecker(max, min, allowzero, psfix)**

Set error = 1

DOWHILE error == 1

Set errno = hasdecimal = haschar = isempty = isoverflow = hasspace = 0

IF psfix == 0 THEN

Display \_string, min, max

ELSE

IF psfix == 1

Display \_string, \_units, min, max

ELSE

IF psfix == 2

Display \_string, min, max, \_units

ELSE

Display \_string, min, max, SPACE, \_units

ENDIF

ENDIF

ENDIF

Call Module fgets(input, 40, stdin)

Set pChar = Call Module strchr(input,’\n’)

IF pChar != 0 THEN

Set length = Call Module strlen(input) – 1

IF input[0] == ‘\n’ THEN

Set isempty = 1

ENDIF

ELSE

IF Call Module fgetc(stdin) == ‘\n’ THEN

Set length = Call Module strlen(input)

ELSE

Set isoverflow = 1

Call Module flush()

ENDIF

ENDIF

IF isoverflow != 1 THEN

DO i = 0 TO i < length

IF Call Module isdigit(input[i]) == 0 THEN

IF i != 0 AND input[i] == ‘.’ AND hasdecimal == 0

Set hasdecimal = 1

continue

ELSE

IF i == 0 AND (input[i] == ‘+’ OR input[i] == ‘-‘) THEN

continue

ELSE

IF input[i] == SPACE OR input[i] == ‘\t’ THEN

Set hasspace = 1

ENDIF

ENDIF

ENDIF

Set haschar = 1

ENDIF

ENDDO

ENDIF

IF isempty == 1 THEN

Display “Detected input is empty, did you key in anything?”

ELSE

IF hasspace == 1 THEN

Display “Space character detected, please try again.”

ELSE

IF isoverflow == 1 THEN

Display “Input exceeds buffer length, please try again.”

ELSE

IF haschar == 0 THEN

Set value = Call Module strtof(input, NULL)

IF value == 0 AND allowzero == 0 THEN

Display “Value detected is 0 and invalid. Please try again.”

ELSE

IF errno == ERANGE OR value < min OR value > max OR value == NAN OR value == INFINITY THEN

Display “Value keyed in is out of range, should be between ‘min’ and ‘max’, please try again.”

ELSE

Set error = 0

ENDIF

ENDIF

ELSE

Display “Character detected, try inputting only numerical values.”

ENDIF

ENDIF

ENDIF

ENDIF

ENDDO

Return value

**END**

**Module dchecker(max, min, allowzero, psfix)**

Set error = 1

DOWHILE error == 1

Set errno = hasdecimal = haschar = isempty = isoverflow = hasspace = 0

IF psfix == 0 THEN

Display \_string, min, max

ELSE

IF psfix == 1

Display \_string, \_units, min, max

ELSE

IF psfix == 2

Display \_string, min, max, \_units

ELSE

Display \_string, min, max, SPACE, \_units

ENDIF

ENDIF

ENDIF

Call Module fgets(input, 40, stdin)

Set pChar = Call Module strchr(input,’\n’)

IF pChar != 0 THEN

Set length = Call Module strlen(input) – 1

IF input[0] == ‘\n’ THEN

Set isempty = 1

ENDIF

ELSE

IF Call Module fgetc(stdin) == ‘\n’ THEN

Set length = Call Module strlen(input)

ELSE

Set isoverflow = 1

Call Module flush()

ENDIF

ENDIF

IF isoverflow != 1 THEN

DO i = 0 TO i < length

IF Call Module isdigit(input[i]) == 0 THEN

IF i != 0 AND input[i] == ‘.’ AND hasdecimal == 0

Set hasdecimal = 1

continue

ELSE

IF i == 0 AND (input[i] == ‘+’ OR input[i] == ‘-‘) THEN

continue

ELSE

IF input[i] == SPACE OR input[i] == ‘\t’ THEN

Set hasspace = 1

ENDIF

ENDIF

ENDIF

Set haschar = 1

ENDIF

ENDDO

ENDIF

IF isempty == 1 THEN

Display “Detected input is empty, did you key in anything?”

ELSE

IF hasspace == 1 THEN

Display “Space character detected, please try again.”

ELSE

IF isoverflow == 1 THEN

Display “Input exceeds buffer length, please try again.”

ELSE

IF haschar == 0 THEN

Set value = Call Module strtod(input, NULL)

IF value == 0 AND allowzero == 0 THEN

Display “Value detected is 0 and invalid. Please try again.”

ELSE

IF errno == ERANGE OR value < min OR value > max OR value == NAN OR value == INFINITY THEN

Display “Value keyed in is out of range, should be between ‘min’ and ‘max’, please try again.”

ELSE

Set error = 0

ENDIF

ENDIF

ELSE

Display “Character detected, try inputting only numerical values.”

ENDIF

ENDIF

ENDIF

ENDIF

ENDDO

Return value

**END**

**Module month\_function(nmonth)**

CASEOF nmonth

1: Set \_cmonth = “Jan\0”

2: Set \_cmonth = “Feb\0”

3: Set \_cmonth = “Mac\0”

4: Set \_cmonth = “Apr\0”

5: Set \_cmonth = “May\0”

6: Set \_cmonth = “Jun\0”

7: Set \_cmonth = “Jul\0”

8: Set \_cmonth = “Aug\0”

9: Set \_cmonth = “Sep\0”

10: Set \_cmonth = “Oct\0”

11: Set \_cmonth = “Nov\0”

12: Set \_cmonth = “Dec\0”

other: return void

ENDCASE

**END**