**Štruktúra WHILE:**

while(podmienka behu)  
{ príkaz 1;    ...    príkaz n;}

Kým bude podmienka behu pravdivá, telo cyklu sa bude opakovať.   
**Príklad 1:**  
#include <stdio.h>  
int main()  
{  
    int i;   
    i = 0;   
    while(i<10) //opakuj kým je i menšie ako 10  
    {   
        printf("%d ", i); //vypíš hodnotu i  
        i++; // inkrementácia  
    };   
    getchar();   
    return 0;  
}  
riadiaca premenná, sa mení v cykle. Podľa hodnoty i cyklus prebehne, alebo ukončí svoju činnosť. Zmena podmienky

**Príklad 2:**  
#include <stdio.h>  
int main()  
{  
    int i;   
    i = 10; //pozor zmena  
    while(i<10) // (10<10) => false = 0  
    {   
        printf("%d ", i);   
        i++;  
    };   
    getchar();   
    return 0;  
}   
Nevypísalo sa nič. Výraz (i<0), podmienka behu, bol vyhodnotený ako nepravdivý (false), preto sa cyklus nevykonal ani raz a bol preskočený.

**DO-WHILE**  
Všeobecná schéma:  
do  
{  
    príkaz 1;  
    ...  
    príkaz n;  
}  
while(podmienka behu);  
  
Cyklus sa vykonáva dovtedy, kým je podmienka behu pravdivá (true), najmenej však jedenkrát.

**Príklad 3:**  
#include<stdio.h>  
int main()  
{  
    int i;  
    i = 10;  
  
    while(i<10) //podmienka vyhodnotená na začiatku cyklu   
    {  
        printf("Som v cykle while. i = %d", i); // toto sa nevypíše  
    };  
    do  
    {  
        printf("Som v cykle do-while. i = %d", i); //toto už áno  
    }  
    while(i<10); //podmienka vyhodnotená na konci cyklu  
  
    getchar();  
    return 0;  
}  
Upozorňujem ešte raz! Cyklus DO-WHILE sa vykoná aspoň raz!!! Drvivá väčšina programátorov využíva len cyklus WHILE, ale občas sa zíde aj DO-WHILE.   
  
Chcete nekonečný cyklus? Nech sa páči!  
  
**Príklad 4 :**  
#include <stdio.h>  
int main()  
{  
    int i=0;   
    while(1) //vždy pravda true = 1  
    {   
        printf("%d ", i);  
        i++;  
    };   
    getchar();   
    return 0;  
}  
alebo  
**Príklad 5:**  
#include <stdio.h>  
int main()  
{  
int i;   
i = 0;   
while(i<10)   
{   
printf("%d ", i);  
// jemne sme niečo zabudli ϑ  
// i sa nebude meniť   
};   
getchar();   
return 0;  
}   
Pozor si dajte na podmienku behu! Taký nekonečný cyklus narobí „zábavu“ v programe.

Pre cykly používame dva príkazy:  
**break** - ukončuje najvnútornejšiu neukončenú slučku cyklu,  
**continue** - spôsobí prechod na ďalší krok cyklu bez toho, aby sa predchádzajúci krok ukončil.

**Príklad 6:**  
#include<stdio.h>  
int main()  
{  
    int i;  
    i = 0;  
    while(1) // nekonečný cyklus  
    {  
        i++;   
        if(i == 7) //ak i =7 vyskočíme z cyklu  
       {  
           printf("nbreak");  
            break; //skoč z cyklu   
        };  
        if(i<10)   
        {  
            printf("nContinue");  
            continue; //pokračuj v cykle  
        };        printf("Toto sa nevypise");   
    };  
    printf("n Nie som v cykle a i=%d",i);  
    getchar();  
    return 0;  
}  
**Príklad na štúdium**  
#include<stdio.h>  
int main()  
{  
    int c;  
    while(1)  
    {  
        c = getchar(); //načítaj znak z obrazovky   
        if(c == 'n') break;  
        putchar(c);  
    };  
    printf("nKoniec");

    getchar();

    return 0;  
}