# Variadic functions

<stdarg.h>

int func Name (data-type variable_name, ...)

int Printf (const char *format, ...);

va_start (va_list ap, argN) → enables access to variadic function

va_arg (va_list ap, type) → accesses next variadic func arg.

va_copy (va_list dest, va_list src) → makes a copy of the variadic func arguments

va_end (va_list ap) → ends traversal of the variadic func. arg.

va_list holds → info needed by start, arg, copy, end.

```
#include <stdarg.h>
#include <stdio.h>

int Add Numbers (int n, ...)
{
   int Sum = 0, i; va_list ptr;
   //Declare pointer to the list pointer
   va_start (ptr, n);
   for (i = 0; i<n; i++)
   {     Sum += va_arg (ptr, int);
   }
   va_end (ptr);
   return Sum;
}
```

Start, get arg!, end

```
int  Sum (int count, ...)
{ int sum=0; [illegible]
  va_list ptr;                      // initialize variable list.
  va_start (ptr, count);
  for (int i=0; i< count; i++)
          Sum+ = va_arg (ptr, int);     // get argument
  va_end (ptr);                     //end va_list pointer.
  return sum;
}
```

```
int min (int count, ...) {
        int min= 100000000, temp;
        va_list ptr;
        va_start (ptr, count);
        for (int i =0; i < ~~10~~ count; i++)
        {
          temp= va_arg(ptr, int);
          if (min > temp)
               min = temp;
          }
        va_end (ptr);
        return min;
}

int    max (int count, ...) {
          int   max =  - 10000000, temp;
          va_list ptr;
          va_start (ptr, count);
          for (int i = 0; i < count; i++)
          {
           temp = va_arg(ptr,int);
           if (max < temp);
                  max = temp;

           }
           va_end (ptr);
           return min;

}
```