

initialization : Exit Controlled Loop;

```

do {
    // code
    // code
    // code
    update();
} while (cond);
```

```

int addMultiple(int a, int b, int m)
{
    int sum = 0;
    while (a >= b)
    {
        if (a % 2 == 0)
            sum += sum + a;
        a--;
    }
    return sum;
}

main() {
    addMultiple(5, 3);
}

```

Ans: 57(3) . Sum = 0, total = 0;

sum = 0%10³

sum = sum + a;

total = total + sum;

a = 5%10³

sum = 5%10³

sum = sum + a;

total = total + sum;

a = 4%10³

sum = 9%10³

sum = sum + a;

total = total + sum;

a = 3%10³

sum = 12%10³

sum = sum + a;

total = total + sum;

a = 2%10³

sum = 14%10³

sum = sum + a;

total = total + sum;

a = 1%10³

sum = 15%10³

sum = sum + a;

total = total + sum;

a = 0%10³

return tot - 15

Diagram illustrating the recursive definition of the sum function:

- Case 1: If $n = 0$, then $\text{sum}(0) = 0$.
- Case 2: If $n > 0$, then $\text{sum}(n) = n + \text{sum}(n-1)$.

```

① Memory Layout
② Storage Classes
③ Section 2: Question
④ Section 4: Question

int subtractProductAndSum(int m) {
    int sum = 0;
    while(m > 0) {
        int digit = m % 10;
        sum += digit;
        prod = prod * digit;
        m = m / 10;
    }
    return prod - sum;
}

```

name: 573
sum: 15
prod: 1

```
Given an integer num, repeatedly add all its digits until the result has only one digit and return it.

Example 1:
Input: num = 38
Output: 2
Explanation: The process is
38 --> 3 + 8 --> 11
11 --> 1 + 1 --> 2
Since 2 has only one digit, return it.
```

$$\begin{aligned} \text{Ans: } & 345 \\ \text{int rev=0;} \\ \rightarrow \text{rem} & = (x \% 10) \\ \text{rev} & = \text{rem} * 10 \\ \text{rev} & = (x \% 10) * 10 \\ \text{C程式: } & (x \% 10) * 10 \end{aligned}$$

- - 9

$$8 \left[\begin{smallmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{smallmatrix} \right] \rightarrow \textcircled{255}$$

$$\text{INT_MIN} \leq \text{INT_MAX} < 999$$

Ans
 $\begin{array}{ccccccccc} O & 1 & 2 & 3 & 5 & 8 & 13 & 21 & 34 \\ 0 & 1 & 2 & 3 & 4 & 6 & 7 & 8 & 9 \end{array}$

```

    int reverse(int n) {
        int rev = 0;
        int n1 = n;
        int nextn = n;
        while (nextn != 0) {
            rev = rev * 10 + nextn % 10;
            nextn = nextn / 10;
        }
        return rev;
    }

```

In C programming, storage classes define the scope, lifetime, visibility, and default initial value of a variable or function. They tell the compiler where the variable will be stored, how long it will exist, and who can access it.

There are four main storage classes in C:

1. Automatic (auto)

- Keyword: auto (but usually omitted, since it's the default for local variables).
- Scope: Local to the block/function in which it is defined.
- Lifetime: Until the end of the block, destroyed when the block is exited.
- Default value: Garbage (undefined).
- Storage: Memory (stack).

Example:

```
void func() {
    auto int x = 10; // same as just "int x = 10";
    printf("%d", x);
}
```

2. Register

- Keyword: register.
- Address: Local to the block/function.
- Lifetime: Same as auto (till function/block execution).
- Default value: Garbage (if available), otherwise RAM.
- Storage: CPU register (if available), otherwise RAM.

Special Note: Address (hex) cannot be taken because it may not be stored in memory.

Example:

```
void func() {
    register int i;
    for (i = 0; i < 10; ++i)
        printf("%d", i);
}
```

3. Static

- Keyword: static.
- Scope: Local to the block defined inside a function, but retains its value between function calls.
- Lifetime: Across multiple function calls.
- Default value: 0 (for integers, NULL for pointers, etc.).
- Storage: Memory (Data segment).

Example:

```
void func() {
    static int count = 0; // initialized only once
    count++;
    printf("func: %d", count);
}
```

4. Extern

- Keyword: extern.
- Address: Visible across multiple files.
- Lifetime: Entire program execution.
- Default value: 0.
- Storage: Memory (Data segment).

Usage: Used to declare a global variable in another file or scope.

Example:

```
// file1.c
int count = 10; // global variable
```

```
// file2.c
extern int count; // declaration, no memory allocated
printf("%d", count);
```

Quick Comparison

Storage Class	Keyword	Scope	Lifetime	Default Value	Storage Location
Automatic	auto	Local (block)	Till block ends	Garbage	Stack
Register	register	Local (block)	Till block ends	Garbage	CPU Register/RAM
Static	static	Local/Global	Entire program	0	Data segment
Extern	extern	Global (file)	Entire program	0	Data segment

Do you want me to also make a **memory layout diagram** (stack, heap, data, code segments) with these storage classes shown visually?

- (1) 509, (left)
- (2) Palindrome 9,
- (3) 208,

509 / 7 / 9 | 250 / 120 / 231

0 1 1 2 3 5 . . .
 formNumber 0 1 2 3 4 5
 (Y) →

```

#include <stdio.h>
int main() {
    int num, reversed = 0, remainder;
    long rev;
    printf("Enter an integer: ");
    scanf("%d", &num);
    while (num != 0) {
        remainder = num % 10; // get last digit
        reversed = reversed * 10 + remainder; // build reversed number
        num = num / 10; // remove last digit
    }
    printf("Reversed number: %d\n", reversed);
    return 0;
}

```

$$\begin{array}{r}
 \underline{\underline{8}} \quad \underline{(123)} \quad \underline{321} \\
 \underline{\underline{16}} \quad \underline{(123)} \quad \underline{321} \\
 \underline{\underline{8}} \quad \underline{8 \cdot 10^0 + 2 \cdot 10^1 + 3 \cdot 10^2} \\
 \underline{\underline{16}} \quad \underline{16 \cdot 10^0 + 2 \cdot 10^1 + 3 \cdot 10^2} \\
 \underline{\underline{8}} \quad \underline{8 \cdot 10^0 + 2 \cdot 10^1 + 3 \cdot 10^2} \\
 \end{array}$$

```

bool isPowerOfTwo(int x) {
    int y;
    long rev = 0;
    if (x == 0) {
        return false;
    }
    if (x < 0) {
        return false;
    }
    if (x < INT_MIN || x > INT_MAX) {
        return false;
    }
    if (x <= 1) {
        return true;
    }
    if (x <= 2) {
        return true;
    }
    if (x <= 4) {
        return true;
    }
    if (x <= 8) {
        return true;
    }
    if (x <= 16) {
        return true;
    }
    if (x <= 32) {
        return true;
    }
    if (x <= 64) {
        return true;
    }
    if (x <= 128) {
        return true;
    }
    if (x <= 256) {
        return true;
    }
    if (x <= 512) {
        return true;
    }
    if (x <= 1024) {
        return true;
    }
    if (x <= 2048) {
        return true;
    }
    if (x <= 4096) {
        return true;
    }
    if (x <= 8192) {
        return true;
    }
    if (x <= 16384) {
        return true;
    }
    if (x <= 32768) {
        return true;
    }
    if (x <= 65536) {
        return true;
    }
    if (x <= 131072) {
        return true;
    }
    if (x <= 262144) {
        return true;
    }
    if (x <= 524288) {
        return true;
    }
    if (x <= 1048576) {
        return true;
    }
    if (x <= 2097152) {
        return true;
    }
    if (x <= 4194304) {
        return true;
    }
    if (x <= 8388608) {
        return true;
    }
    if (x <= 16777216) {
        return true;
    }
    if (x <= 33554432) {
        return true;
    }
    if (x <= 67108864) {
        return true;
    }
    if (x <= 134217728) {
        return true;
    }
    if (x <= 268435456) {
        return true;
    }
    if (x <= 536870912) {
        return true;
    }
    if (x <= 1073741824) {
        return true;
    }
    if (x <= 2147483648) {
        return true;
    }
    if (x <= 4294967296) {
        return true;
    }
    if (x <= 8589934592) {
        return true;
    }
    if (x <= 17179869184) {
        return true;
    }
    if (x <= 34359738368) {
        return true;
    }
    if (x <= 68719476736) {
        return true;
    }
    if (x <= 137438953472) {
        return true;
    }
    if (x <= 274877906944) {
        return true;
    }
    if (x <= 549755813888) {
        return true;
    }
    if (x <= 1099511627776) {
        return true;
    }
    if (x <= 2199023255552) {
        return true;
    }
    if (x <= 4398046511104) {
        return true;
    }
    if (x <= 8796093022208) {
        return true;
    }
    if (x <= 17592186044416) {
        return true;
    }
    if (x <= 35184372088832) {
        return true;
    }
    if (x <= 70368744177664) {
        return true;
    }
    if (x <= 140737488355328) {
        return true;
    }
    if (x <= 281474976710656) {
        return true;
    }
    if (x <= 562949953421312) {
        return true;
    }
    if (x <= 1125899906842624) {
        return true;
    }
    if (x <= 2251799813685248) {
        return true;
    }
    if (x <= 4503599627370496) {
        return true;
    }
    if (x <= 9007199254740992) {
        return true;
    }
    if (x <= 18014398509481984) {
        return true;
    }
    if (x <= 36028797018963968) {
        return true;
    }
    if (x <= 72057594037927936) {
        return true;
    }
    if (x <= 14411518807585968) {
        return true;
    }
    if (x <= 28823037615171936) {
        return true;
    }
    if (x <= 57646075230343872) {
        return true;
    }
    if (x <= 115292150460687744) {
        return true;
    }
    if (x <= 230584300921375488) {
        return true;
    }
    if (x <= 461168601842750976) {
        return true;
    }
    if (x <= 922337203685501952) {
        return true;
    }
    if (x <= 184467440737000384) {
        return true;
    }
    if (x <= 368934881474000768) {
        return true;
    }
    if (x <= 737869762948001536) {
        return true;
    }
    if (x <= 1475739525896003072) {
        return true;
    }
    if (x <= 2951479051792006144) {
        return true;
    }
    if (x <= 5902958103584012288) {
        return true;
    }
    if (x <= 11805916207168025576) {
        return true;
    }
    if (x <= 23611832414336051152) {
        return true;
    }
    if (x <= 47223664828672102304) {
        return true;
    }
    if (x <= 94447329657344204608) {
        return true;
    }
    if (x <= 18889465931468809216) {
        return true;
    }
    if (x <= 37778931862937618432) {
        return true;
    }
    if (x <= 75557863725875236864) {
        return true;
    }
    if (x <= 15111572745175073376) {
        return true;
    }
    if (x <= 30223145490350146752) {
        return true;
    }
    if (x <= 60446290980700293504) {
        return true;
    }
    if (x <= 120892581961400587008) {
        return true;
    }
    if (x <= 241785163922801174016) {
        return true;
    }
    if (x <= 483570327845602348032) {
        return true;
    }
    if (x <= 967140655691204696064) {
        return true;
    }
    if (x <= 1934281311382409392128) {
        return true;
    }
    if (x <= 3868562622764818784256) {
        return true;
    }
    if (x <= 7737125245529637568512) {
        return true;
    }
    if (x <= 15474250491059275136024) {
        return true;
    }
    if (x <= 30948500982118550272048) {
        return true;
    }
    if (x <= 61897001964237100544096) {
        return true;
    }
    if (x <= 123794003928474201088192) {
        return true;
    }
    if (x <= 247588007856948402176384) {
        return true;
    }
    if (x <= 495176015713896804352768) {
        return true;
    }
    if (x <= 990352031427793608705536) {
        return true;
    }
    if (x <= 1980704062855587217411072) {
        return true;
    }
    if (x <= 3961408125711174434822144) {
        return true;
    }
    if (x <= 7922816251422348869644288) {
        return true;
    }
    if (x <= 15845632522844697739288576) {
        return true;
    }
    if (x <= 31691265045689395478577152) {
        return true;
    }
    if (x <= 63382530091378790957154304) {
        return true;
    }
    if (x <= 12676506018275758191430864) {
        return true;
    }
    if (x <= 25353012036551516382861728) {
        return true;
    }
    if (x <= 50706024073103032765723456) {
        return true;
    }
    if (x <= 101412048146206065531446912) {
        return true;
    }
    if (x <= 202824096292412131062893824) {
        return true;
    }
    if (x <= 405648192584824262125787648) {
        return true;
    }
    if (x <= 811296385169648524251575296) {
        return true;
    }
    if (x <= 1622592770339297048503150592) {
        return true;
    }
    if (x <= 3245185540678594097006301184) {
        return true;
    }
    if (x <= 6490371081357188194012602368) {
        return true;
    }
    if (x <= 12980742162714376388025204736) {
        return true;
    }
    if (x <= 25961484325428752776050409472) {
        return true;
    }
    if (x <= 51922968650857505552100818944) {
        return true;
    }
    if (x <= 103845937301715011104201637888) {
        return true;
    }
    if (x <= 207691874603430022208403275776) {
        return true;
    }
    if (x <= 415383749206860044416806551552) {
        return true;
    }
    if (x <= 830767498413720088833613103056) {
        return true;
    }
    if (x <= 1661534996827440177667226206112) {
        return true;
    }
    if (x <= 3323069993654880355334452412224) {
        return true;
    }
    if (x <= 6646139987309760710668904824448) {
        return true;
    }
    if (x <= 13292279974619521421337809648896) {
        return true;
    }
    if (x <= 26584559949239042842675619297792) {
        return true;
    }
    if (x <= 53169119898478085685351238595584) {
        return true;
    }
    if (x <= 106338239796956171370702477191168) {
        return true;
    }
    if (x <= 212676479593912342741404954382336) {
        return true;
    }
    if (x <= 425352959187824685482809858764672) {
        return true;
    }
    if (x <= 850705918375649370965619717529344) {
        return true;
    }
    if (x <= 1701411836751298741931239435058688) {
        return true;
    }
    if (x <= 3402823673502597483862478870117376) {
        return true;
    }
    if (x <= 6805647347005194967724957740234752) {
        return true;
    }
    if (x <= 1361129469401038993544985548046904) {
        return true;
    }
    if (x <= 2722258938802077987089971096093808) {
        return true;
    }
    if (x <= 5444517877604155974179942192187616) {
        return true;
    }
    if (x <= 1088903575520831194835984438375232) {
        return true;
    }
    if (x <= 2177807151041662389671968876750464) {
        return true;
    }
    if (x <= 4355614302083324779343937753500928) {
        return true;
    }
    if (x <= 8711228604166649558687875507001856) {
        return true;
    }
    if (x <= 1742245720833329911737575101400372) {
        return true;
    }
    if (x <= 3484491441666659823475150202800744) {
        return true;
    }
    if (x <= 6968982883333319646950300405601488) {
        return true;
    }
    if (x <= 1393796576666663929380060081120296) {
        return true;
    }
    if (x <= 2787593153333327858760120162240592) {
        return true;
    }
    if (x <= 5575186306666655717520240324481184) {
        return true;
    }
    if (x <= 1115037261333331143504048064896232) {
        return true;
    }
    if (x <= 2230074522666662287008096129792464) {
        return true;
    }
    if (x <= 4460149045333324574016192259584928) {
        return true;
    }
    if (x <= 8920298090666649148032384519169856) {
        return true;
    }
    if (x <= 17840596181333281496064769038339712) {
        return true;
    }
    if (x <= 35681192362666562992129538076679424) {
        return true;
    }
    if (x <= 71362384725333125984259076153358848) {
        return true;
    }
    if (x <= 142724769450666259768518152306717696) {
        return true;
    }
    if (x <= 285449538901332519536036304613435392) {
        return true;
    }
    if (x <= 570898577802665039072072609226870784) {
        return true;
    }
    if (x <= 1141797156053310078144145218453741568) {
        return true;
    }
    if (x <= 2283594312106620156288290436907483136) {
        return true;
    }
    if (x <= 4567188624213240312576580873814966272) {
        return true;
    }
    if (x <= 9134377248426480625153161747629932544) {
        return true;
    }
    if (x <= 18268754496852961250306323495259865088) {
        return true;
    }
    if (x <= 36537508993705922500612646990519730176) {
        return true;
    }
    if (x <= 73075017987411845001253293981039460352) {
        return true;
    }
    if (x <= 146150035974223690025066587962078920704) {
        return true;
    }
    if (x <= 292300071948447380050133175924157841408) {
        return true;
    }
    if (x <= 584600143896894760100266351848315682816) {
        return true;
    }
    if (x <= 1169200287933789520200532703696631365632) {
        return true;
    }
    if (x <= 2338400575867579040400166407393262731264) {
        return true;
    }
    if (x <= 4676801151735158080800332814786525462528) {
        return true;
    }
    if (x <= 9353602303470316161600665629573050925556) {
        return true;
    }
    if (x <= 18707204606940632323201331259146101851112) {
        return true;
    }
    if (x <= 3741440921388126464640266251829220370224) {
        return true;
    }
    if (x <= 7482881842776252929280532503658440740448) {
        return true;
    }
    if (x <= 14965763685532555858560665007316881480896) {
        return true;
    }
    if (x <= 29931527371065111717121330014633762961792) {
        return true;
    }
    if (x <= 59863054742130223434242660029267525923584) {
        return true;
    }
    if (x <= 119726109484260446668485320058535051847168) {
        return true;
    }
    if (x <= 239452218968520893336970640117070103694336) {
        return true;
    }
    if (x <= 478904437937041786673941280234140207388672) {
        return true;
    }
    if (x <= 957808875874083573347882560468280414777344) {
        return true;
    }
    if (x <= 1915617751748167146695765120936560829544688) {
        return true;
    }
    if (x <= 3831235503496334293391530241873121658089376) {
        return true;
    }
    if (x <= 7662471006992668586783060483746243316178552) {
        return true;
    }
    if (x <= 15324942013985337173566120967492486632357004) {
        return true;
    }
    if (x <= 30649884027970674347132241934984973264714008) {
        return true;
    }
    if (x <= 61299768055941348694264483869969946529428016) {
        return true;
    }
    if (x <= 122599536111882897388528967739939893058856032) {
        return true;
    }
    if (x <= 245199072223765794777057935479879786117712064) {
        return true;
    }
    if (x <= 490398144447531589554115870959759572235424128) {
        return true;
    }
    if (x <= 980796288895063179108231741919519144470848256) {
        return true;
    }
    if (x <= 1961592577790126358216463483838593889016965112) {
        return true;
    }
    if (x <= 3923185155580252716432926967677187778033930224) {
        return true;
    }
    if (x <= 7846370311160505432865853935354375556067860448) {
        return true;
    }
    if (x <= 1569274062232101086571770787070751111213572096) {
        return true;
    }
    if (x <= 3138548124464202173143541574141502222427444192) {
        return true;
    }
    if (x <= 6277096248928404346287083148283004444854883984) {
        return true;
    }
    if (x <= 1255419249785680869257176629656600888909767976) {
        return true;
    }
    if (x <= 2510838499571361738514353259313201777819535952) {
        return true;
    }
    if (x <= 5021676999142723477028706518626403555639071904) {
        return true;
    }
    if (x <= 1004335399828546695405741303725280711127853808) {
        return true;
    }
    if (x <= 2008670799657093390811482607450561422255707616) {
        return true;
    }
    if (x <= 4017341599314186781622965214901122844511415232) {
        return true;
    }
    if (x <= 8034683198628373563245930429802245689022830464) {
        return true;
    }
    if (x <= 16069366397256747126498608
```