**HR Question**

## Express myself.

## Explain about my major and mini project in UG.

## Why I need to hire you?

## How you are connected with society?

## Why do you think TCS is your best choice?

## How can you serve society through your job in TCS?

## What are the products of TCS?

## What are TCS’s new projects?

## Are you ready to reallocate?

## So will you stay in TCS for more than 10/5 years?

## What are your other activities?

## Do you want to do further studies?

## What is your vision?

## What do you know about TCS-vision employee’s revenue (some facts and figures)?

## If you get selected, Will you choose those projects in TCS?

## 

## Technical Question

## What is difference between Class and Object?

## What is tuple?

## What are attribute?

## What is agile?

## Explain object oriented concept?

## What are the software models?

## What is prototype model?

## How many normal forms are there?

## What is 1NF, 3NF?

## Show how we delete intermediate node in singly linked list and doubly linked list?

## What is your favorite subject?

## What is different between C and C++?

## What is different between structure and union?

## What is different between structure and array?

## What is %c and %d in C?

## What is pointer?

## What is size of long and double and a pointer variable?

## Write a program to add 2 numbers?

## Write a program to reverse the string?

## Do you know anything about bond?

## What is difference between Null pointer and Zero Pointer?

## Write a program for string palindrome checking?

## Do you have any questions to ask us?

## Write a program which run C code without main().

## 1) Using a macro that defines main

|  |
| --- |
| #include<stdio.h>  #define fun main  int fun(void)  {      printf("running C code without main");      return 0;  } |

Output:

running C code without main

**2) Using Token-Pasting Operator**  
The above solution has word ‘main’ in it. If we are not allowed to even write main, we can use token-pasting operator (see [this](http://www.geeksforgeeks.org/interesting-facts-preprocessors-c/) for details)

|  |
| --- |
| #include<stdio.h>  #define fun m##a##i##n  int fun()  {      printf("running C code without main ");      return 0;  } |

Output:

running C code without main ]

**To find sum of two numbers without using any operator**

Write a C program to find sum of positive integers without using any operator. Only use of printf() is allowed. No other library function can be used.

|  |
| --- |
| int add(int x, int y)  {      return printf("%\*c%\*c",  x, ' ',  y, ' ');  }    int main()  {      printf("Sum = %d", add(3, 4));      return 0;  } |

Output:

Sum = 7

The output is seven spaces followed by “Sum = 7″. We can avoid the leading spaces by using carriage return.

|  |
| --- |
| int add(int x, int y)  {      return printf("%\*c%\*c",  x, '\r',  y, '\r');  }    int main()  {      printf("Sum = %d", add(3, 4));      return 0;  } |

Output:

Sum = 7

**How can we sum the digits of a given number in single statement?**

Below are the solutions to get sum of the digits.

**1. Iterative:**  
The function has three lines instead of one line but it calculates sum in line. It can be made one line function if we pass pointer to sum.

|  |
| --- |
| # include<stdio.h>  int main()  {    int n = 687;    printf(" %d ", getSum(n));      getchar();    return 0;  }    /\* Function to get sum of digits \*/  int getSum(int n)  {      int sum;      /\*Single line that calculates sum\*/      for(sum=0; n > 0; sum+=n%10,n/=10);      return sum;  } |

**2. Recursive**  
Thanks to ayesha for providing the below recursive solution.

|  |
| --- |
| int sumDigits(int no)  {    return no == 0 ? 0 : no%10 + sumDigits(no/10) ;  }    int main(void)  {    printf("%d", sumDigits(1352));    getchar();    return 0;  } |

**Calculate Logn in one line**

Write a one line C function that calculates and returns \lfloor\log _2 \left( n \right)\rfloor. For example, if n = 64, then your function should return 6, and if n = 129, then your function should return 7.

Following is a solution using recursion.

|  |
| --- |
| #include<stdio.h>    unsigned int Log2n(unsigned int n)  {     return (n > 1)? 1 + Log2n(n/2): 0;  }    int main()  {    unsigned int n = 32;    printf("%u", Log2n(n));    getchar();    return 0;  } |

Let us try an extended verision of the problem. Write a one line function Logn(n ,r) which returns \lfloor\log _r \left( n \right)\rfloor. Following is the solution for the extended problem.

|  |
| --- |
| #include<stdio.h>    unsigned int Logn(unsigned int n, unsigned int r)  {     return (n > r-1)? 1 + Logn(n/r, r): 0;  }    int main()  {    unsigned int n = 256;    unsigned int r = 4;    printf("%u", Logn(n, r));    getchar();    return 0;  } |

Below is another tricky code can be used to print “Even” or “Odd” accordingly. Thanks to [student](http://www.geeksforgeeks.org/archives/8337/comment-page-1#comment-4987) for suggesting this method.

|  |
| --- |
| #include<stdio.h>  int main()  {      int no;      printf("Enter a no: ");      scanf("%d", &no);      (no & 1 && printf("odd"))|| printf("even");      return 0;  } |

**How will you print numbers from 1 to 100 without using loop?**

Here is a solution that prints numbers using recursion.  
  
Other alternatives for loop statements are recursion and goto statement, but use of goto is not suggestible as a general programming practice as goto statement changes the normal program execution sequence and makes it difficult to undestand and maintain.

|  |
| --- |
| #include <stdio.h>    /\* Prints numbers from 1 to n \*/  void printNos(unsigned int n)  {    if(n > 0)    {      printNos(n-1);      printf("%d ",  n);    }    return;  }    /\* Driver program to test printNos \*/  int main()  {    printNos(100);    getchar();    return 0;  } |

**Write a C program to print “Great job” without using a semicolon**

Use printf statement inside the if condition

|  |
| --- |
| #include<stdio.h>  int main()  {        if( printf( "Great job " ) )        {    }  } |

One trivial extension of the above problem: Write a C program to print “;” without using a semicolon

|  |
| --- |
| #include<stdio.h>  int main()  {     if(printf("%c",59))     {     }  } |

## Write a one line C function to round floating point numbers

**Algorithm:** roundNo(num)  
1. If num is positive then add 0.5.  
2. Else subtract 0.5.  
3. Type cast the result to int and return.

**Example:**  
num = 1.67, (int) num + 0.5 = (int)2.17 = 2  
num = -1.67, (int) num – 0.5 = -(int)2.17 = -2  
 **Implementation:**

/\* Program for rounding floating point numbers \*/

# include<stdio.h>

int roundNo(float num)

{

return num < 0 ? num - 0.5 : num + 0.5;

}

int main()

{

printf("%d", roundNo(-1.777));

getchar();

return 0;

}

Output: -2

**Implement Your Own sizeof**

Here is an implementation.

|  |
| --- |
| #define my\_sizeof(type) (char \*)(&type+1)-(char\*)(&type)  int main()  {      double x;      printf("%d", my\_sizeof(x));      getchar();      return 0;  } |

You can also implement using function instead of macro, but function implementation cannot be done in C as C doesn’t support function overloading and sizeof() is supposed to receive parameters of all data types.

**How to find length of a string without string.h and loop in C?**

Find the length of a string without using any loops and string.h in C. Your program is supposed to behave in following way:

Enter a string: hello (Say user enters hello)

Entered string is: hello

Length is: 5

You may assume that the length of entered string is always less than 100.

The following is solution.

|  |
| --- |
| #include <stdio.h>    int main()  {       char str[100];       printf("Enter a string: ");       printf( "\rLength is: %d",                printf("Entered string is: %s\n", gets(str)) - 20             );         return 0;  } |

Output:

Enter a string: hello

Entered string is: hello

Length is: 5

**What is NULL pointer?**   
**Ans:** NULL is used to indicate that the pointer doesn’t point to a valid location. Ideally, we should initialize pointers as NULL if we don’t know their value at the time of declaration. Also, we should make a pointer NULL when memory pointed by it is deallocated in the middle of a program.

**What is Dangling pointer?**  
**Ans:** Dangling Pointer is a pointer that doesn’t point to a valid memory location. Dangling pointers arise when an object is deleted or deallocated, without modifying the value of the pointer, so that the pointer still points to the memory location of the deallocated memory. Following are examples.

|  |
| --- |
| // EXAMPLE 1  int \*ptr = (int \*)malloc(sizeof(int));  .............  .............  free(ptr);    // ptr is a dangling pointer now and operations like following are invalid  \*ptr = 10;  // or printf("%d", \*ptr); |
| // EXAMPLE 2  int \*ptr = NULL  {     int x  = 10;     ptr = &x;  }  // x goes out of scope and memory allocated to x is free now.  // So ptr is a dangling pointer now. |

**What is memory leak? Why it should be avoided**  
**Ans:** Memory leak occurs when programmers create a memory in heap and forget to delete it. Memory leaks are particularly serious issues for programs like daemons and servers which by definition never terminate.

|  |
| --- |
| /\* Function with memory leak \*/  #include <stdlib.h>    void f()  {     int \*ptr = (int \*) malloc(sizeof(int));       /\* Do some work \*/       return; /\* Return without freeing ptr\*/  } |

**What is l-value?**  
l-value or location value refers to an expression that can be used on left side of assignment operator. For example in expression “a = 3″, a is l-value and 3 is r-value.  
l-values are of two types:  
“nonmodifiable l-value” represent a l-value that can not be modified. const variables are “nonmodifiable l-value”.  
“modifiable l-value” represent a l-value that can be modified.

**How to write your own sizeof operator?**

|  |
| --- |
| #define my\_sizeof(type) (char \*)(&type+1)-(char\*)(&type) |

# What's the difference between a null pointer and a void pointer?

Null pointer is a special reserved **value** of a pointer. A pointer of any type has such a reserved value. Formally, each specific pointer type (int \*, char \* etc.) has its own dedicated null-pointer value. Conceptually, when a pointer has that null value it is not pointing anywhere.

Void pointer is a specific pointer **type** - void \* - a pointer that points to some data location in storage, which doesn't have any specific type.

So, once again, null pointer is a **value**, while void pointer is a **type**. These concepts are totally different and non-comparable. That essentially means that your question, as stated, is not exactly valid. It is like asking, for example, "What is the difference between a triangle and a car?".

**2.**      **How to measure the size of any variable without “sizeof” operator?**

**Answer:**

#define   size\_of(x)   ((char \*)(&x+1) – (char \*)&x)

void main()

{

Short int x;

Printf(“%d”,size\_of(x));

}

8.      **Test whether a no. is power of 2 or not.**

**Answer:**

void main ()

{

int n;

printf (“\n Enter any no:”);

scanf (“%d”, & n);

if (n & & ((n & n-1) = = 0))

    printf (“It is power of 2”);

else

   printf (“It is not power of 2”);

}

**Test:**

Enter any no: 32

It is power of 2

Enter any no: 56

It is not power of 2

10.  **Write a C-program which does the addition of two integers without using ‘+’ operator.**

**Answer:**

Logic-1:

c = a – (-b);

**Q Where was the first potato found?**

**Ans: In the ground.**

**Q: What comes down but never goes up?**

**Ans: rain.**

**Q: If three cats kill three rats in three minutes, how long will it take hundred cats to kill hundred rats?**

**Ans: three minutes.**

**Q: What can fly but has no wings?**

**Ans: Time.   
Q: I m like a ribbon, tied by nature, across the sky, what m I?**

**Ans: Rainbow.**

**Q: How would u write nineteen that if one is taken out, then its remains twenty.**

**Ans: XIX when one is taken out, its remains XX.**

**Q: There were ten sparrows sitting on a tree. A hunter fired and two of them fell dead. How many sparrows were left on the tree?**

**Ans: Non.**

**Q: Two sons and two fathers went hunting. They succeeded in hunting one pigeon each on counting it was found that they were only three pigeons. How is that?**

**Ans: They were only three persons, son father and grandfather.**

**Q: which part of London in France?**

**Ans: -N-**

**Q: why ur nose is not twelve inches long?**

**Ans: Because then it would be a foot.**

**Q: What r the largest ant in the world?**

**Ans: Elephant.**

**Q: what is the easiest way to get to heaven quickly?**

**Ans: just stand in front of the fast moving car**

**Q: Where do fish keep their money?**

**Ans: at the river bank.**

**Q: Which sea has waves but no water?**

**Ans: BBC.**

**Q: Which is the most shocking city in the world?**

**Ans: electricity.**

**Q: Which fish lives in heaven?**

**Ans: Angel fish.**