

.....pointers.....

1)what is the difference between c and emnbedded c?

A)embedded c is an extension of the c language.

normal c is used in the application oriented programming where as embedded c is used in the microcontroller related application (or) boards

c is used in desktop applications and embedded c is used in microcontroller or processor or sensor based applications.

1)what is dangling pointer?

A)it is a pointer in which its points to the memory location at that memory location there is no valid data .

then its called as dangling pointer. the pointer which is points to the deallocated memory location is called dangling pointer.

after free function we can assign the null to the pointer then it will be called as null pointer.

2)what is null pointer?

A) the pointer which is assigns or initilized with null value is called null pointer.

```
ex.int *ptr=NULL;
```

3)what is void pointer?

A)the pointer which can be declared of type void and used to store any of the data type by typcast to corresponding data type.then it will be called

as void pointer.it will be called as generic pointer ie it holds the addresses of any type by simply type cast it so it will be called as generic pointer.

```
ex..void *ptr;  
(int*)ptr=10;  
(char*)ptr='c';
```

4)what is wild pointer?

A)the pointer declared and it will be not initialized so it will be called as wild pointer.it is having wild behavior because of not initialized.

```
ex. int *ptr;
```

5)what is near pointer?

A)the pointer is pointing with in the segment then it is called as near pointer.

this type of pointer is having 16 bytes of range.

6)what is far pointer?

A)a far pointer is pointing to the outside of the segment is called far poinetr. and

this pointer is also having 16 byte segment. and it will also points to the out side of the segment .

7)what is Double pointer?

A) the pointer which stores the address of the another pointer variable is called as double pointer

or

the pointer is pointing to the another pointer address then it is called as double pointer.

```
ex. int x=10
    int *ptr=&x;
    int **ptr=&ptr;
```

usage used in dynamic memory allocations in linked list, functions.

8) explain about pointer to constant and constant pointer?

A) pointer to constant: you can not modified the data pointed by the pointer .but you can change the pointer to point different data.

syntax: `const int *ptr;`

constant pointer: you can modify the value pointed by the pointer but you cannot change the pointer to point to different data.

syntax: `int *const ptr;`

constant pointer to a constant: in this you cannot change the data pointed by pointer and you cannot change the pointer to point different data.

syntax: `const int *const ptr;`

9) what are the operation we cannot perform on pointers?

A) 1) addition, multiplication and division operations are not performed on two pointers.

2) multiplication between pointer and any number.

3) division of a pointer by any number.

4) addition of float and double values of a pointers.

10) what are the operations can be performed on pointers?

A)

1) increment and decrement operations are performed.

2) subtraction of two pointers.

3) all comparison operators operations can be performed on pointers.

11) explain about array of pointer and pointer to an array?

A) array of pointer: an array of pointer is essentially an array but the array elements are of type pointer. this is useful to store multiple addresses

of same data type.

syntax: `int* ptr[];`

ex: `int* ptr[5];`

pointer to an array: it is a pointer which is pointing to the whole array elements rather than single element.

syntax: `int (*ptr)[];`

ex: `int arr[10]={1,2,3,4,5};`

`int (*ptr)[10]=&arr;`

12) what is the size of the pointer?

A) 4 bytes in 32 bit machine.

8 bytes in 64 bit machine.

13) where pointer is stored?

A) pointer is declared as locally then it is stored in stack.

if it is declared as globally then it is data or bss. if you declare a pointer globally most of the time it will be act as a dangling pointer means

if we creat globally then the life time is throughout the program then if it is free and we cannot assign this as null then it leads to dangling pointer.

14) what is pointer?

A) pointer is a variable which stores the address of the another variable of any type

15) what is array of pointer and pointer to an array?

A) pointer to an array: the pointer is points to the whole array elements then it is called as array of pointer.

syntax is : data type (*pointer)[sizeof the array];

array of pointer: the pointer in which the array of elements itself a pointer then it is called as array of pointer. or in an array the elements of type is of pointer type then it is called as array of pointer.

syntax: data type *ptr[elements];

16) differences between arrays and pointers?

A) arrays: 1) the array name it self is a constant pointer and it gives the base address of the array or it points to the first element of the array.

2) array is having a fixed name we cannot modify the name of the array.

3) size of the array is also fixed.

4) arrays are stored in stack or heap.

pointers: 1) pointer points to the memory address of the another variable.

2) pointers can points the different memory locations.

17) pointers are integers or not?

A) no.. pointers are not integers but it will be represents in the integer way. like 32 bit 64 bit way.

18) what is function pointer?

A) the pointer is pointing to the addresss of the function then it is called as function pointer.

syntax of the function pointer is : (void*) ptr(actual parameters separated by comma);

int* ptr(int a, int b) = &func;

19) what is file pointer?

A) the pointer of type file which points to the addresss of the file variable then it will points to the file is called as file pointer.

20)if i create a pointer in locally with static type then where it will be stored?

A)if we declare a pointer with static key word then it will be stored in data or bss.

```
int *globalPtr; // Stored in global/static memory
void example()
{
    static int *staticPtr; // Also stored in global/static memory
}
```

.....structures.....

1)what is meant by user defined data type?

A) user will create the data type to store the variables is called as user defined data type.

there are 3 types of user defined data types

- 1)structures
- 2)union
- 3)enumeration

2)what is structure and explain brief?

A)structure is a user defined data type.it will store the heterogenous data items into one location.

```
syntax:struct name
{
    member 1;
    member 2;
    and so on;
}varname;
```

structure is creating memory when we declare a structure variable.otherwise it will not creating memory.

3)what are the possible ways of creating a structure variables?

A)there are two ways.

1)at the time of structure creation

```
struct name
{
    member 1;
    member 2;
    and so on;
}varname;
```

2)

normal initialization

```
struct name
{
    member 1;
    member 2;
    and so on;
};
struct name varname;
```

4)what is self referential structure?

A) in the structure at least one member is of type pointer and it points to the

same type of the structure is called self referential structure.

```
ex: struct name
    {
        int a;
        struct name *ptr;
    }varname;
```

5) what is structure padding?

A) in structure members we have one of the member is having the large data type and

that data type is assigned to all the remaining structure members is called as structure padding.

```
ex: struct name
    {
        int a; //8bits
        char b; //it uses the above empty bits
        float d; //it allocates 8bits
        double c; //it allocates 8 bits
    }varname;
```

6) how to avoid structure padding?

A) by using `#pragma pack(1)`

7) what is the difference between structure and array?

A) 1) structure is used to store heterogeneous type of data items.

2) structures are used to perform on linked list operations.

3) it is user defined data type.

arrays: 1) array is predefined data type.

2) array stores homogeneous data items.

3) arrays are used in the repetitive sections.

8) calculate the size of the structure?

```
A) struct name
    {
        int a; //8bytes
        char b; //it uses the above empty bits
        float d; //it allocates 8bytes
        double c; //it allocates 8 bytes
    }varname;
```

the size of the above structure is 24 bytes.

9) can a structure contain a pointer to itself?

A) yes that is called as self referential structure .and it is used to dynamically create a linked list ,trees,graphs.

10) where structures are stored?

A) structures memory allocation generally with `malloc` calls so it will go and stored in heap.

11) Why can't you initialize a data element of a structure inside a structure?

A). In a structure, the data elements have no physical meaning. Memory in a structure is allocated only after the variable (object) associated with it is created. The data members of a structure are accessed through variables (objects) and hence initializing a value to a data member inside a structure is absurd.

.....unions.....

1) what is union?

A) union is a user defined data type same as structure but the difference is structure allocates memory for all the members of the structure. but in union the largest data type of the union member is only having the memory.

2) what is the output of the code?

```
union person{
    int age;
    char name[20];
}person1;

int main()
{
    person1.age=24;
    strcpy(person1.name, "praveen");
    printf("%d\n", person1.age);
    printf("%s", person1.name);
}
```

3) differences between structure and union?

A)

structures:

- 1) structures are user defined data type.
- 2) structures will allocate memory for all the members at a time with structure padding concept.
- 3) basically structures are used in linked list.

union:

- 1) union is also user defined data type.
- 2) union will allocate memory only for the highest data type member only so if you can perform only one operation at a time. otherwise memory overlap will happen.

4) unions allow padding or not?

A)unions doesnt allow the padding .it will take the highest data type and perform the task so here no padding concept is occur.

.....storage classes.....

1)what are the types of storage classes?

A)mainly there are four types of storage classes they are

- 1)auto storage class
- 2)extern storage class.
- 3)static storage class.
- 4)register storage class.

2)explain about auto storage class?

A)auto storage class is the general storage class or a default storage class.

lifetime of this auto storage class is with in a block.

scope is also with in the block.

it has max stored in stack

if you are only declaring then it will be having garbage values.

3)explain about register storage class?

A)register storage class is used to store the register values.

it will be stored in cpu registers for faster access rather than cache memory.

it will also having same as auto storage scope ,life time,initialized value

except storage.

by using any special calls we cannot change its scope.

Q)can i print the registers address?

A)no we didnt print the registers address.it gives the compilation error.

4)explain about extern storage class?

A)extern storage class is used to access one variable defined in one file and access in another

file then we will use extern storage class

it has global scope

life time is also global

storage is not created it will access to another file so if we create a variable

then it will store at particular segment then it will be stored in that segment only

5)explain static storage class?

A)static storage class is used in several ways as declared as it with global variables ,local variables and functions.

it is having file scope via external linkage and it is used with another file also we can access it depends on the conditions.

it is having life time as end of the file

it is have been stored in data or bss

6) can i apply static to functions?

A) yes you can apply static keyword to function.

7) what is the return type of size of operator and where it will be defined?

A) the return type of size of operator is `size_t`.

and its definition is present in the `stddef.h` or `stdint.h`

8) without using `sizeof` operator can we get the sizes of `int`, `char`, `arr[]`?

A) yes we can get the sizes of those data type variables.

ex1) for integer:

```
int main()
{
    int x=10;
    int *ptr=&x;
    int *ptr2=ptr+1;
    int size=(char*)ptr2-(char*)ptr1;
    printf("%d\n",size);
}
```

```
ex2) char n='c';
    char *ptr=&n;
    char *ptr2=ptr+1;
    int size=ptr2-ptr;
    printf("%d",size);
```

```
ex3)    int arr[10];
        int size=*&arr+1)-arr;
        printf("%d",size);
```

9) what is storage classes?

A) storage class defines the variables scope, visibility, where it will be store and what is the initial values in the process or in the function.

10) what is cache memory and types of cache memories?

A) cache memory is the temporary memory which is used to store the repeatedly used variables or functions for saving the time.

cache memory is in between the hard disk and ram.

types of cache : l1 cache: it is placed in the cpu registers. and it is the faster memory compared to the remaining two.

l2 cache: it placed near to the cpu registers and it is slow compared to l1.

l3 cache: it is placed far away from cpu registers and it is slower than the two caches.

11) what is fragmentation ? explain its types?

A) wastage of memory in memory segments are nothing but fragmentation . there are two types of fragmentation is occurs there are

1. internal fragmentation
2. external fragmentation

internal fragmentation:if the allocated memory section is not filled sufficiently it will be having some empty memory is there in the memory section then the unused memory section is called internal fragmentation.

external fragmentation:if the allocated memory is sufficiently filled and still some memory wants to storing the remaining instructions .so this will be called as external fragmentation.

disadvantages of fragmentation:1)system performance will be decreased
2)disk will be having some extra space will be not filled.

.....dynamic memory allocation.....

1)what are the ways to allocate memory for variables?

A)two ways to allocate memory for variables and they are

1)static memory allocation.

2)dynamic memory allocation.

static memory allocation:when we allocate the memory in program time or at compile time is called as static memory allocation .

if we allocate memory at compile time they will store in data,bss and stack.depending on the initialization.

```
ex:  int a=10;
      or  int a;
      scanf("%d",&a);
```

dynamic memory allocation:when we allocate memory at run time then it is called as dynamic memory allocation .

we can allocate memory dynamically in 3 ways:

1)using malloc();

2)using calloc();

3)realloc();

4)free();

if you allocate memory dynamically it will stored in heap segment.

malloc():it is one way to allocate memory dynamically.

syntax: void *ptr=(void*)malloc(sizeof (data type));

after creating the memory it will not happen any initialization.

it returns on success a pointer to be pointed to the block of memory and it will be of void* type and

we convert it into any type by type cast to it.

on failure it returns NULL.

malloc internally invokes the brk system call to allocate the smaller blocks of memory.

brk() system call is used to allocate the small amount of memory .it will be increases the heap area.

mmap is also a system call which is used to allocate the large amount of memory.

calloc():

 syntax: void *ptr=(void*)calloc(no of elements,sizeof(data type));

 after successfully creating the memory it will be initialize with zeros.

 these zeros are filled by the internal loop to set each byte to zero or memset() function is filling the zeros

 on success calloc returns same as the malloc

 on failure it will return NULL

 it will also internally invoke the brk mmap system calls same as malloc.

 it allocates the memory block by block.

 memset is a C standard library function which is used to set a particular value to the block of created memory.

 syntax: memset(address of the created memory, int what value you want to set, size_t number of bytes)

realloc():

 it will reallocate that means already allocated memory by using malloc or calloc then it will be reallocate the size of the memory.

 syntax: void *ptr2=(void*)realloc(void*ptr,sizeof(data type));

 so it will be having two parameters.

 on success it will return the pointer points to the newly created memory.

 on failure it will return the NULL.

free() : if you allocate the memory then it will be created and after it will be used then you must deallocate the memory otherwise it leads to be memory leak.

 so free function is used to free or delete the dynamically allocated memory.

 free does not return anything.

4) malloc and calloc which one is faster?

A) compare to calloc malloc is faster because malloc does not initialize the created memory

 and calloc creates memory and after that it will initialize the memory with zeros.

 and malloc has continuous memory allocation but calloc will be having block by block allocation.

 so malloc is faster.

5) diff between malloc and calloc?

A) malloc contains only one argument

 malloc is faster

 it creates a continuous memory.

 it does not initialize the created memory

 malloc is faster.

calloc having two arguments

calloc is slower than the malloc because it is initialized with zeros.

calloc also allocates the non continuous memory allocation with number of blocks.
after creating the memory it will be initialized with zero.
calloc is slower compared to malloc.

6) what is memory leak and how can we avoid the memory leaks?

A)if we dynamically allocate the memory for variables after the usage of the variables we cannot free or release the memory then memory leaks are happen.

if we can free the memory using the free function then we can avoid memory leaks.

7)how to check the memory leaks are happen?

A) 1)by using mtrace and muntrace

2)by using valgrind method

valgrind --leak-check=full --show-leak-kinds=all ./memory_leak

3)using clang address-sanitize we will use where the memory is leaked this info it will be displayed

8)why free(ptr) is deleting the all created memory through dma?

A)if we create the memory through malloc we will create the memory in run time

these call will make a linked list internally .

and also it will be creating the some extra memory for the meta data and the end of the data block corresponding address.

so this will be stored in the extra bytes created by the dma calls

so when ever we can free the memory then in the pointer starting address to the end of the list will be pointed so that allocated whole block will be deallocated.

9)how much amount of extra memory is created for the meta data and the pointer storage purpose?

A)in 32 bit it will create extra over head of 16 bytes.

64 -bit it will creat extra over head of 24 to the 32 bytes

in that size_t size for 8,

for pointer 8bytes

for int free 4 bytes

placeholder val[1].

10)

.....recursion.....

1)what is recursion?and what is the advantages of recursion?

A)recursion is nothing but the function which is calling itself is called as recursion.

advantages of recursion:1)code optimisation.
2)code reusability.
3)memory wastage reduced.

2)what are the types of recursion?

A)there are four types of recursions are there

1)direct recursion:the function is calling directly itself is called direct recursio

2)indirect recursion:in this in one function called as another helper function to calling the function is called as indirect recursion.

3)tail recursion: last operation is the recursive call operation then it is called as tail recursion.

4)non_tail recursion:last call is not the recursive function call it may be having some operation is performed then this type of recursion is called as non tail recursion.

3)what are the phases of recursion?

A)there are two phases 1)winding phase:in this phase recursive calls are made until it reaches to base case.

here stack frames are created.

2)unwinding phase:after reaching the base casse of recursive function then unwindind phase is started.

int this we will returning the values and deallocate the memory for stack frames.

4)can i return more than one value to a function or not?

A)yes you can return more than one value to a function by using the call by reference and structure pointer.

ex:

```
int main()
{
int a=10,b=20,sum,diff,mul;
func(a,b,&sum,&diff,&mul);
printf("%d\n%d\n%d\n",&sum,&diff,&mul);
}
int func(int a,int b,int *sum,int *diff,int *mul)
{
*sum=a+b;
*diff=a-b;
*mul=a*b;
}
```

5)what is function returning pointer?

A)a function that returns the pointer is called as function returning pointer.

6)what is function pointer?

A)a pointer that points to the functions base address then it is called as function pointer.

ex:

```
int main()
{
int a=1,b=2;
```

```

int *ptr(int,int);
//func(a,b);
ptr=&func;
printf("%d\n",ptr);
}
int func(int a,int b)
{
int c=a+b;
return c;
}

```

7)where we will use function pointers and advantages?

A)function pointers can be efficiently used in apis and libraries.
via call back functions.

advantages:code reusability,less memory space,rwadability.

8)what is call back function?

A)a call back function is nothing but in function call one of the
argument is also

a function call then it is called call back function

```

int func1(int a,int b)
{
int c=a+b;
return c;
}
int func2(int a,int b)
{
int d=a-b;
return d;
}

```

```

ex:
int main()
{
int a,=2,b=4;
func(a,b,func1(add));
//func(a,b,func2(sub));
}
int func(int a,int b,void*callback(x)(int,int))
{a
printf("%d%d",a,b);
callback(a,b);
}

```

9)what is the function prototype in c?

A)function declaration is called as function prototype.

10)what are the ways to pass the function arguments?

A)1)call by value->directly values as arguments.

2)call by reference->value address is passed as arguments to the
function

so in function deinition we will use pointer to receive the addresses.

11)what are the types of functions?

A) predefined ex: `printf()`, `scanf()`, `gets()`, `puts()`, `getc()` and `putc()`.
user defined function : these functions are created by programmer for their usage.

12) what are builtin functions?

A) builtin functions are nothing but predefined functions.

13) how to get the functions address?

A) by using function name we can get the functions address.

14) where functions are stored ?

A) whole function is stored in text segment in the form of binary instructions.

after that it will be stored in stack.

15) what is the difference between `gets()` and `fgets()`?

A) if you use `gets()` and `fgets()` both are used to take input as a string from the user.

`gets()`: in `gets()` if you didn't mention the size of the character array size so the condition is to take that boundary

but by using the `gets()` we can take input from the user beyond the boundary.

so it leads to runtime errors.

syntax: `gets(string name or base address of the string);`

`fgets()`: if you use `fgets()` to avoid the `gets()` drawback.

`fgets()` gives the input a string from the user with the addition of newline character (`\n`).

syntax: `fgets(string name, sizeof(string name), stdin);`

16) what is macro and usage of macros in our program?

A) macros are nothing but symbolic constants which are executed before main function.

macros are having several advantages.

1) by using macros we can remove the complications and our code will be easily readable and writable

2) if we can modify the values in program in normal way we can go to every time where it is used and modified it but by

using macros we can change the value at once your entire code is replaced with the macro.

17) why will we use `sprintf()` and `sscanf()`?

A) `sprintf()` is used to print any integer to string.

syntax: `sprintf(str, %d, x);`

`sscanf()`: it is used to give the standard input and it gives any string to integer.

syntax: `sscanf(str, %d, &x)`

18) what is endianness? explain the types of endianness?

A)endianess is used to how the data is stored in memory.
there are two types of endian ness we have

1)big endian:in this the msb position bit will be stored in the first memory segment or block.it will be called as big endian.

ex :12 34 56 78

big endian: 12 34 56 78 .

2)littel endian:in this the lsb position bit will be stored in the statrting bits of the memory block then it is called as littel endian.

ex:12 34 56 78

littel endian:78 56 34 12.

to check system is which type of endian ness is explained by byte checking or byte array checking.

19)what is stack over flow?

A)stack is used to creating the memory for local variables and formal arguments.

when the memory allocated for stack area is 2MB.if this memory range will be crossed then the stack is filled and it will go for another segment

so it will give the segmentation fault with stack over flow error.

causes of stack over flow:

1)deep recursion or infinete recursion.

2)large local variables.

avoid stack over flow:to avoid stack over flow we want to take return statments and dont do deep recursion.

20)what is typedef and what is the use of it?

A)typedef is used to give aliasis for the complex names or commonly used name.it will give the code more readable and maintainable.

so by using typedef the complex names will be make simple and readable.

21)what is volatile keyword in c? and where it will be used mostly?

A)it is used to get the updated variable value at any time.

whenever the compiler see the volatile keyword then the compiler will not apply the compiler optimization and it will give the updated values.

for example we use a volatile keyword with a variable and the variable stores the sensor reading values then it will go through the sensor hardware registers and take the accurate or updated values.

1)hardware registers :for knowing status of the hardware value.

2)shared memory:it will be used if two threads shared a same variable then one thread update the value and another thread is read.

3)interrupt service routine.

22)what is catchee memory?

A) cache memory is a temporary memory. it will store the repeated variables info in a process is used the those variables repeatedly used in code.

it will give faster access.

cache memory is present in between cpu and ram.

it is costliest memory.

typically it will be of size 2mb.

23) difference between cache memory and registers?

A) registers memory is very fast compared to cache memory.

cache memory is costlier than registers

registers are present in cpu.

cache memory is present in between the cpu and ram.

25) what is conditional compilation?

A) the compiler is include or exclude the parts of the code based on certain condition.

based on the this it will be compiling the code.

in our cgrof project we are used the conditional compilation to the LTE modem for the communication over INDIA and to USA in this switching will be used through conditional compilation.

26) swap char* ptr="praveen";
char arr[]="sanju";?

A)

```
char arr1[20];  
strcpy(arr1, arr);  
strcpy(arr, ptr);  
ptr = arr;  
printf("%s\n%s\n", ptr, arr);
```

27) what is the difference between gets and fgets?

A) gets and fgets both are used to take input as string from the user.

in gets only one argument is there that is address of the string.

but in fgets there are three arguments 1) string address

2) sizeof the string.

3) stdin

syntax of fgets : fgets(base address of the string, strlen(string), stdin);

syntax of gets : gets(string);

the main difference is in gets is it will not check the boundary, user give large amount of input it will take after running it gives segmentation fault.

in fgets it will check the boundary and give input at that boundary only.

28) what is memcpy and memmov?

A) memcpy is a standard library function to copy the specified number of bytes from one memory location to another memory location.

it will be having the issue as overlapping of the data.

syntax. memcpy(dest, source, strlen(source string)+1);

memmov:the disadvantage of the memcpy is overcome by using the memmov

29)what is the difference between memcpy and memmove?

A)1)memcpy is used to copy the specified bytes to the another memory location.

2)it is very efficient way to copy.

3)but it is having a problem with overlapping.

4)it will be faster than memmove.

memmove:1)memmove is used to change the bits of the same memory location.

2)so it will be not much an efficient way .

3)it will be not having any overlapping issue.

4)it will be slower compared to memcpy.

30)what is the difference between memcpy and strcpy?

A)memcpy:1)it is used to copy the specified bytes of memory for one memory location to another memory location.

2)it will be working with any type of data.

3)it will be having an issue with overlapping of the strings.

strcpy:it is used to copy the whole string excluding null character.

2)it will be applicable only with strings.

31)any number divid by 4 simple one line answer?

A) $x=x>>2$.

32)what is function and what are their advantages?

A)function is a block of code executed sequentially.

advantages:code reusability.

decrease the memory

modularity

creation of libraries by using functions gives less stress.

creating debugging tools

33)whata is the purpose of using main function?

A)main function is the entry point of the function

for the main execution will be started.

main function is organizes the function calls structures.

it will be handling the command line arguments for providing input at run time through terminal.

main function will give the return value as 0 to the operating system then the program executoin will be successfully done.

34)what is an argument and types of arguments ?

A)arguments are nothing but the values passed to the functions.

there are two types of arguments

1)actual arguments:it will be mention in function call then it is called as actual arguments.

2)formal arguments:which is mentioned in function definition.

35) what is builtin function?

A) the function is a predefined function and which are not read or whole function definition it can be readily available to perform the related operations.

36) When does the compiler not implicitly generate the address of the first element of an array?

A) by using the address operator to the array then it will give whole array.

in the case of size of operator then it will give the whole array size not first element.

during the indexing values means arr[3] gives 4th element not first element.

37) Write the equivalent expression for x%8?

A) x&7.

38) why n++ executes faster or n+1?

A) n+1 is executed faster because of these conditions

1) n++ will be first change the value means increment and then it prints the old value

n+1 it will perform simply perform addition.

2) in assembly level understanding n++ is perform read store and write will be applied but n+1 it will perform just addition so n+1 will be executed faster.

39) difference between #include< > & #include " ".

A) #include<>: it will be the pre-defined header file for the execution of the code in compile time

<> these angle brackets tell the compiler to see the system folders of these directory or these header file is present or not.

it is mainly used to include the predefined headers in the system folders.

include"" : it will be the user-defined headers.

"" these quotes indicate to the compiler that first check the current directory if no header is present then you will go to the system folders.

it is mainly used in creating our own headers in the project.

40) what is checksum error?

A) the error occurs at the time of transmission of data or storing of data at the time the error will occur then it is called checksum error. reasons

data corruption, hardware failure, software interrupts.

41) multiply by 9 give the fastest way to give the logic?

A) `int n=18;`
 `int s=(n<<3)+n;`

42) Which has greater time complexity: Iteration or Recursion and Why?

Ans. Recursion has a greater time complexity as the entire function is called repeatedly whereas in iteration, the time complexity simply depends on the number of cycles of the loop. Internally, for every recursion, a new stack frame will be created in the computer memory.

so that's why recursion is time complexity high

44) how do you debug the code manually?

A) by using `printf` statement printing each functionality and behaviour of the code. then we can understand the flow of the code.

by using tools we can also debug by using `gdb`, `jlink`.

45) what are the modifications that will happen in linking stage of the compilation stages?

A) in linking stage if the main function is present or not.

- 2) linking of libraries.
- 3) adding some functionalities to the compiler.
- 4) it creates the executable file.
- 5) it creates the memory segments.

46) Discuss the Role of Preprocessor Directives in C.

A) Preprocessor directives in C programming instruct the compiler to process them before compiling the code.

Common directives include `#include`, `#define`, and `#ifdef`. highlighting their role in code inclusion, macro definitions, and conditional compilation.

50) 0. What are macros? What are its advantages and disadvantages?

Ans: Macros are abbreviations for lengthy and frequently used statements.

When a macro is

called the entire code is substituted by a single line though the macro definition is of several lines.

The advantage of macro is that it reduces the time taken for control transfer as in case of

function. The disadvantage of it is here the entire code is substituted so the program becomes lengthy if a macro is called several times.

51) `bss` stands for?

A)block started by symbol

52)what is the difference between printf and puts ?

A)puts is used to print the string type only .

but coming to printf it will print any type of variable by using the formatt specifier.

53)where file pointers and file variables are stored?

A)it is created statically it will be present in the stack
file variables are stored in heap.

54)what is fseek?

A)it is used to tell about to move the cursor position of file pointer.
it will be having 3 arguments

fseek(fp,offset,position);

in offset where do you move the cursor

coming to position it will having 3 types

seek_set:it sets the where you want to set the file pointer

seek_cur: it tells about the file pointers current position

seek_end: it will give the function pointer to the end of the file.

55)what is ftell() and what is the purpose?

A)it is a function used to tells about the cursor position.as well as it tells the length of the file.

56)what is rewind()?

A)it is a function is used to where ever the cursor position of the function pointer then it will be shifting to the first letter of the file.

57)what is file?and what is file pointer?

A)a file is storing the sequential data perminently in the hard disk.it will be having the sequential bytes of data.

file pointer is the pointer is pointing to the file type of variable is called as file pointer from this file type variable it will be having the actual file address.

58)modulus operator returns what?

A)modulus operator returns integer only not floating if you trying to do operation with modulus with floating point value you facing with compilation error.

if you want to perform the floating point value with modulus then we need the fmod function which is defined in math.h library function.
fmod takes two double values then it performs the mdulus operation then give the remainder.

59)without using modulus operator you can get the remainder?

A)we have the bulit in function fmod(double,double)

60)why sizeof operator is not a function?

A)sizeof is a operator not a function because it is not occupying the memory in text segment and function is occupying the memory

sizeof is a compiled time operator.

sizeof is not occupying even its parameters but functions occupying the memory for their parameters

61)what is ram dump?

A)it is the process of storing the computers ram memory into the secondary storage devices like hard disk.

this saves when the system is crashes then the information so it is safely saves the memory.

62)what is static library and what is dynamic library?differences?

A)static library:

1)it is a collection of object files created by the user and it will be linked at compiled time then it is called as static library.

2)first to create the static library we will be extracting the .o file from .c file and then we will create the static library file with the .o file

to create the static library we have the command `ar rcs lib(name of the .a file) .o file name`

3)then we will compile the main.c file with the created library file for linking the static library file we will use the command

`gcc main.c -L. -l(static library file name without .a extension)`

dynamic library:

1) dynamic libraries are loaded at run time and it will consuming the memory only at once and later it will be reuse the memory.

example is `libc.so` for all c standard library calls.

2)the dynamic libraries are extensions with the .so name.

3)it will also same as static library creation but at .o file generation we will give `gpic` command to generate shared object.

63)which looping is fastest ?

A)do while loop is faster than the remaining loops

if the remaining two loops will check the condition then it is success then it will go and execute the loop part

but do while loop first it will go to the loop and then it checks the condition.

64)differences between for and while loops?

A)for loop:

if the number of iterations will be known then we are using the for loop.

for loop is the entry controlled loop

it is easy to understand and debug.

while loop:
if the number of iterations will be unknown then we will using the while loop
it is also called as exit controlled loop
it is difficult to understand and debug

65)what are the differences between the sizeof and strlen function?

A)sizeof():it is the compile time operator.

it didnt occupy any memory because the sizeof operator work will be done

during compilation stage so no memory is occupied.

it will calculate how much memory occupied by the array or any variable

it will calculate the including the special character.

strlen: it is a runtime function.

it will counts the character array size.

it will not calculate the termination character.

it will be used only for the strings.

66)what is inline and outline functions in c?

A)inline function:if the function defination contains the keyword inline before the function.

then the function call will be replaced by the function definition or function body.

if multiple times we can invoke the function then after every function call compiler will go and search for the function definition

if we use the inline keyword then it will be replace the function call with the function definition and access fastly.

ex :

main()

{

int a=10,b=20;

func(a,b);//the function call will be replaced by the function definition

func(a,b);

inline void func(int a,int b)

{

int c=a+b;

printf("%d\n",c);

}

}

outline function:

if the normal function calling is the outline function.

it will be having the normal behavior.it is slower process compared to the inline function.