

Q1. What is the output of fun(25)? (2 points)

```
void fun(int n){  
    if (n==0)  
        return ;  
    fun(n/2);  
    printf("%d",n%2);  
}
```

A. 10011

B. 11001

C. 11101

D. 10111

Answer: 11001

Q2. (Find the Output) (4 marks)

What is the output of the code given below? (4 marks)

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

```
void foo(int *x, int* y){  
    int *temp = x;  
    x = y;  
    y = temp;  
    *x = *y + 7;  
    *y = 7;  
}  
int main(){  
    int a = 5;  
    int b = 5;  
    foo(&a, &b);  
    printf("%d %d", a, b);  
    return 0;  
}
```

Answer: 7 12

Q3. (Multi Correct) (3 marks = 0.75x4)

We are given a static multidimensional array "int M[4][4]". Which of the following is(are) the correct way(s) to get the value at M[i][j] (where i and j are valid indices)? (3 marks = 0.5x6)

*(&M[0][0] + 4*i + j) (Correct)
*(&M[0][0] + 4*j + i)
((M+i)+j) (Correct)
((M+j)+i)
*(&M[i][0]+j) (Correct)
*(M[i]+j) (Correct)

Q4. (FIB) (3 Marks)

Fill in the blanks so that the code outputs "25" (without quotes) with atleast 2 calls to the "helper" function. (3 marks)

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

```
int helper(int x){  
    if(x==1){  
        return ____swers;  
    }  
    return x+helper(x-____);  
}  
int main(){  
    printf("%d", helper(____));  
    return 0;  
}
```

Answer: 1, 2, 9 OR 0, 24, 25

Q5. (Find the Output) (2 Marks) **(Bonus, everyone will get marks)**

Write the output of this code given that you know that it compiles correctly?

```
#include <stdio.h>

int main(){
    char str[] = "ESC IS BEST";
    int *a = (int*)str;
    printf("%s", (char *)(a+1));
    return 0;
}
```

Answer: "IS BEST" (without quotes)

Q6. (5 Marks) (1+2+2)

```
Line 0 int trimulti(int n){
Line 1     if (n==2) return 2;
Line 2     if (n==1) return 1;
Line 3     else if (n==0) return 0;
Line 4     else return trimulti(n-1)*trimulti(n-3)*trimulti(n-4);}
```

How many times tri multi calls itself recursively when trimulti(5) is executed?

Answer: infinitely

This function was supposed to calculate trimulti-nacci number which is governed by the expression : $f(n)=f(n-1)*f(n-2)*f(n-3)$ starting with terms 1,1,2 where 1 is the zeroeth term. Select lines to be changed to make trimulti function to execute properly with the changed line.

Answer:

```
Line 3 : else if (n==0) return 1;
Line 4: else return trimulti(n-1)*trimulti(n-2)*trimulti(n-3);
```

How many times tri multi calls itself recursively when trimulti(5) is executed with the corrected code.

Answer: 12

Q7. Complete the following program. (1x8=8 marks.)

[For the first blank, 1 mark should be given to all as a bonus]

Given a sequence of characters(only a to z letters are allowed), for each character find the nearest character to its right having lower ASCII value than current character. If there is no such character then the character will be '#'.

For example, if the input consists of the characters "dcgfegibdc", then the output of the program should be "c#febib#c#" (without quotes).

```
void modifySequence(char a[]){
    int len = sizeof(<1>);
    char temp1,temp2;
    int j,i;
    for(i=0;i < <2>;i++)
    {
        for(j=i+1;j < <3>;j++)
        {
            if(<4>)
            {a[i]=a[j];
              <5>
            }
        }
        if(<6>)a[i]='#';
    }
    return ;
}

int main() {
    char arr[];
    printf("Input the Sequence\n");
    scanf(<7>);

    modifySequence(<8>);

    printf("Printing Modified Sequence\n%s",arr);
    return 0;
}
```

Answer: a,len-1,len,a[i]>a[j],break,j==len,("%s",arr),arr

Q8. Multi-correct (3*1 = 3 marks)

We have,

```
float a = 1.55;
```

```
float *b = &a;
```

Which of the following options will give 1 as output.

A. `printf("%d", **b);`

B. `printf("%d", (int)'9' - (int)'8');`

C. `printf("%ld", (b + 1) - b);`

D. `printf("%d", (int)*(&a));`

Answer: B,C,D

Q9. How many different integer values can variable x be such that output of program is 2020. (3 Marks)

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

```
void foo(int a){  
    if(a % 10 == 0) printf("%d", a);  
    else foo(a-1);  
}  
int main(){  
    int x;  
    foo(x);  
    return 0;  
}
```

Answer: 10

Q10. FIB (2 marks)

Fill in the blank if the output of following program is 6,

```
int arr[] = {1, 4, 9, 16, 25};  
int *a = &arr[0];  
int*b = a + ____;  
int x = (*b - *a)/(b - a);  
printf("%d", x);
```

Answer: 4

Q11 (Marks 6 = 1.5x4)

You are given the following incomplete code to compute and print the base k notation of a given decimal number.

Fill the blanks appropriately.

```
void convert(int n, int k)
{
    if(**(1)**)
    {
        convert(**(2)**, **(3)**);
        printf("%d", **(4)**);
    }
}
```

For example, if convert(6,2) is called, it should print 110 as output.

Answer:

1 : n > 0

2 : n / k

3 : k

4 : n % k

Q12 - How many times will the recurse function be called? (6 Marks)

...

```
#include<stdio.h>
int value ;
int count =1;
void recurse( int array[] , int sz )
{
    printf("Times = %d", count);
    count++;
    if ( sz == 1 )
        return ;
    array [ sz / 2] = value ;
    value ++;
    recurse ( array , sz /2 ) ;
    recurse ( array + sz /2 , sz /2 ) ;
}
int main ()
{
    value = 0 ;
    int sz = 5 ;
```

```

int array [ 5 ] ;
for ( int i = 0 ; i < sz ; ++i )
{
    array [ i ] = 200+ i ;
}
recurse ( array , sz ) ;
for ( int i = 0 ; i < sz ; ++i )
{
    printf ( "%d\n" , array [ i ] ) ;
}
return 0;
}
...

```

Answer: 7

Q13. Find the output. (5 Marks)

```

...
#include<stdio.h>
int main() {
    int arr[] = {10, 20, 30, 40, 50, 60};
    int *ptr1 = arr;
    int *ptr2 = &*(arr + 3) + 2;
    printf("%d\n", (ptr2 - ptr1));
    printf("%d\n", (char*)ptr2 - (char*)ptr1);
    printf("%d %d %d", sizeof(ptr1), sizeof(ptr2), sizeof(arr));
    return 0;
}
...

```

Answer:

5
20
8 8 24

Q14 (7 marks = 5 + 2)

See the code snippet below and answer the questions.

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

void bar(int *pa, int b, int c);

void foo(int *pa, int b, int c)
{
    if(__BLANK_1__)
        return;
    else if(b == 0)
        bar(pa, b, c);
    else
    {
        __BLANK_2__;
        bar(pa, __BLANK_3__, c);
    }
}

void bar(int *pa, int b, int c)
{
    if(__BLANK_4__)
        return;
    else if(c == 0)
        foo(pa, b, c);
    else
    {
        *pa = *pa * 3;
        __BLANK_5__;
    }
}

int main()
{
    int n = 1;

    foo(&n, 5, 2);
    printf("%d ", n);

    n = 1;
    bar(&n, 5, 2);
}
```



```

        printf("%d", n);

        return 0;
    }

    ...

```

Part 1

Both `foo()` and `bar()` are should multiply 2^{a3^b} to the value pointed by pointer `pa`. Fill in the blanks so that the code works correctly.

Answers:

- * Blank 1: `b == 0 && c == 0`
- * Blank 2: `*pa = *pa * 2`
- * Blank 3: `b-1`
- * Blank 4: `b == 0 && c == 0`
- * Blank 5: `foo(pa, b, c-1)`

Part 2

How many times `foo` and `bar` will be called respectively? `foo` = _____, `bar` = _____.

Answers:

10 and 11

Q15 (2 marks)

Select the expressions which will give address of the variable a?:

```

``c
int main()
{
    int a = 10;
    printf("%x", __expression_for_address_of_a__);
    return 0;
}

...

```

1. `&a`
2. `&(*(&a + 1))-1`
3. `*&*&*&a`
4. `&*&*&*&a`

Answers: 1, 2, 4

Q16 (2 marks)

What is the output of the program given below?

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

int flag = 1;

int main(int argc){
    if(flag){
        flag = 0;
        printf("pow(2,%d): %d\n", -7, main(-7));
        return 0;
    }
    if(!argc)
        return 1;

    int pow_half = main(argc/2)*main(argc/2);
    if(argc%2)
        pow_half = pow_half*2;

    return pow_half;
}
...

```

Options:

1. `pow(2, -7): 0`
2. `pow(2, -7): 128`
3. Compilation Error
4. It will compile but gives Runtime Error because of the recursive call to main

Answer: Option2

Q17 (2 Marks)

You are given the following program. You are supposed to find the output of the program.

...

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

```
void swapPointers(int *a, int *b){
```

```
int * temp = a;
```

```
a = b;
```

```
b = temp;
```

```
}
```

```
int main() {
```

```
int a[10];
```

```
for(int i=0;i<10;i++){
```

```
a[i] = i;
```

```
}
```

```
int *p = a+2;
```

```
int *q = a+8;
```

```
swapPointers(p, q);
```

```
printf("%d", p-q);
```

```
}
```

...

Answer: -6

Find the output of the following code:

...

- 1) 71
- 2) 33 (**CORRECT**)
- 3) 11
- 4) Compilation Error
- 5) Runtime Error

...

Let a be a 2d matrix which is defined as `int a[7][7];`. Which of the following are same as `a[5][6]`?

- (1) ``*(a+5) + 6`
`
- (2) ``**[a[5]+6]`
`
- (3) ``*(a[5]+6)`
`
- (4) ``*(a+6))[5]`
`
- (5) ``*(a+5*4+6)`
`
- (6) ``**[a+5*4+6]`
`
- (7) ``*(a+5*4+6)`
`

Answer: 3

Q20 (1 marks)

The valid dynamic allocation of an array of pointer in C:

- i. ``int **mat = (int **) malloc(N *sizeof(int*))``
- ii. ``int **mat = (int **)calloc (N,sizeof(int))``
- iii. ``int ** mat = (int **)malloc(N*sizeof(int))``
- iv. ``int ** mat = (int *)malloc(N*sizeof(int*))``

Answer:

- i. ``int **mat = (int **) malloc(N *sizeof(int*))``

Q21 (1 marks)

In theory, sizeof can return both positive and negative value.

Answer: False

Q22 (1 marks)

The main function cannot call itself recursively.

Answer: False

Q23 (1 marks)

Consider following lines of code

```
int *x = 5;  
int **y = &x;
```

Size of x and y will be the same.

Answer: True

Q24 (1 marks)

Passing an array to a function is exactly the same as passing the pointer to the first element of the array.

Answer: True

Q25

Multidimensional arrays cannot be initialized in declarations like 1-D arrays.

Answer: False