TAKE HOME ASSIGNMENT-04
Deep Dive into Pointers
Practical Programming with C (CSE 3544)

Problem Statement:

Working with pointers, pointer to arrays, array of pointers, function returning pointer, pointer to function.

Do I Able to Answer? Evaluate:

1. Consider the following ANSI C program;

```c
#include<stdio.h>
int main(){
  int arr[4][5],i,j;
  for(i=0;i<4;i++){
    for(j=0;j<5;j++){
      arr[i][j]=10*i+j;
    }
  }
  printf("%d ",arr[2][4]);
  printf("%d\n",*(*(arr+2)+4));
  return 0;
}
```

What is the output of the above program?

Output with explanation	
(A) 14 14	(C) 24 24
(B) 24 14	(D) 14 24

[GATE 2021]

2. Consider the following ANSI C program;

```c
#include<stdio.h>
int main(){
  int arr[4][5];
  int i,j;
  for(i=0;i<4;i++){
    for(j=0;j<5;j++){
      arr[i][j]=10*i+j;
    }
  }
  printf("%d\n",*(arr[1]+9));
  return 0;
}
```

What is the output of the above program?

Output with explanation	
(A) 14	(C) 24
(B) 20	(D) 30

3. What is printed by the following ANSI C program?

[GATE 2022]

```c
#include<stdio.h>
int main(void)
{
  int x = 1, z[2] = {10, 11};
  int *p = NULL;
  p = &x;
  *p = 10;
  p = &z[1];
  *(&z[0] + 1) += 3;
  printf("%d, %d, %d\n", x, z[0], z[1]);
  return 0;
}
```

Output with explanation	
(A) 1, 10, 11	(C) 10, 14, 11
(B) 1, 10, 14	(D) 10, 10, 14

4. What is printed by the following ANSI C program?   **[GATE 2022]**
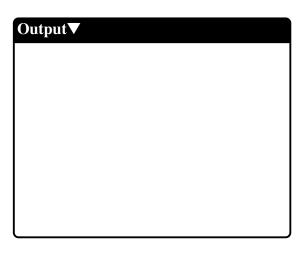
```c
#include<stdio.h>
int main(int argc, char *argv[])
{
 int a[3][3][3] =
 {{1, 2, 3, 4, 5, 6, 7, 8, 9},
 {10, 11, 12, 13, 14, 15, 16, 17, 18},
 {19, 20, 21, 22, 23, 24, 25, 26, 27}};
 int i = 0, j = 0, k = 0;
 for( i = 0; i < 3; i++ ){
   for(k = 0; k < 3; k++ )
      printf("%d ", a[i][j][k]);
   printf("\n");
 }
 return 0;
}
```

Output▼

5. What is printed by the following ANSI C program?   **[GATE 2022]**

```c
#include<stdio.h>
int main(int argc, char *argv[])
{
 int a[3][3][3] =
 {{1, 2, 3, 4, 5, 6, 7, 8, 9},
 {10, 11, 12, 13, 14, 15, 16, 17, 18},
 {19, 20, 21, 22, 23, 24, 25, 26, 27}};
 int i = 0, j = 0, k = 0;
 for( i = 0; i < 3; i++ ){
   for(k = 0; k < 3; k++ )
      printf("%d ", a[i][j][k]);
   printf("\n");
 }
 return 0;
}
```

Output▼

6. Consider the program below;   **[GATE 2009]**

```c
#include<stdio.h>
int fun(int n, int *f_p){
  int t,f;
  if(n<=1){
    *f_p=1;
    return 1;
  }
  t=fun(n-1, f_p);
  f=t+ *f_p;
  *f_p=t;
  return f;
}
```

```c
int main(){
 int x=15;
 printf("%d\n",fun(5,&x));
 return 0;
}
```

The value printed is

Output▼

(A) 6	(C) 14
(B) 8	(D) 15

7. Consider the following C program   **[GATE 2020]**

```c
#include<stdio.h>
int main()
{
  int a[4][5]={
        {1,2,3,4,5},
        {6,7,8,9,10},
        {11,12,13,14,15},
        {16,17,18,19,20}};
  printf("%d\n",*(*(a+**a+2)+3));
  return 0;
}
```

The output of the program is

Output▼

8. Consider the following C function;   **[GATE 2020]**

```c
int tob(int b, int *arr){
  int i;
  for(i=0;b>0;i++){
    if(b%2)
        arr[i]=1;
    else
        arr[i]=0;
    b=b/2;
  }
  return(i);
}
```

```c
int pp(int a, int b){
  int arr[20];
  int i,tot=1,ex,len;
  ex=a;
  len=tob(b,arr);
  for(i=0;i<len;i++){
   if(arr[i]==1){
        tot=tot*ex;
   ex=ex*ex;
   return(tot);
}
```

The value returned by **pp(3,4)** is _____

Write the execution pattern and final output▼

  (i) **len:**

  (ii) **arr content:**

  (iii) **tot:**

  (iv) **ex:**

        Finally, **pp(3,4)**:

9. Write the output of the following program;

```c
#include<stdio.h>
void fun(int *,int *);
int main()
{
  int i=5,j=5;
  fun(&i,&j);
  printf("%d  %d\n",i,j);
  return 0;
}
```

```c
void fun(int *p,int *q){
  p=q;
  *q=10;
}
```

Output▼

10. Find the output and study the ways addresses are getting displayed;

```c
int main(){
    int a[]={12,13,14,15,16,17};
    printf("%p, %p\n",a,&a,&a[0],&a[0]+2);
    printf("%u, %u\n",a,&a,&a[0],&a[0]+2);
    printf("%d, %d\n",a,&a,&a[0],&a[0]+2);
    printf("%p, %p\n",&a[0],&a[0]+2);
    printf("%u, %u\n",&a[0],&a[0]+2);
    printf("%d, %d\n",&a[0],&a[0]+2);
    return 0;
}
```

Output▼

11. State the output of the following program. Assume the address of p is 1000 and q is 2000.
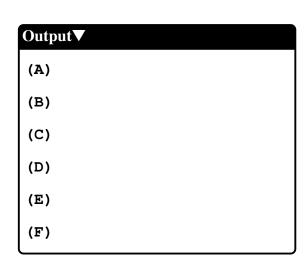
```c
#include<stdio.h>
void fun(int **q);
int main(){
    int x;
    int *p=&x;
    *p=55;
    fun(&p);
    printf("%d %p\n",*p,p);
    return 0;
}
```

```c
void fun(int **q){
    int r=20;
    **q=r;
    printf("%p\n",*q);
}
```

Output▼

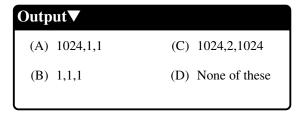12. Write the output of the code snippet by observing the co-relation of pointer manipulation in 2-D array.

```c
int main(){
    int n=4,m=3;
    int a[n][m];
    int (*p)[m]=a;
    p=p+1;
    (*)p[2]=100;
    n=p-a;
    printf("%d\n",n);           /*----(A) */
    printf("%d\n",(*p)[2]);     /*----(B) */
    printf("%d\n",*((*p)+2));   /*----(C) */
    printf("%d\n",*(a[1]+2));   /*----(D) */
    printf("%d\n",*(*p+2));     /*----(E) */
    printf("%d\n",*(p[0]+2));   /*----(F) */
    return 0;
}
```

Output▼

(A)

(B)

(C)

(D)

(E)

(F)

13. Select the output of the following program.

```c
int main(){
    int a[][3]={4,5,6,7,8,9,1,2,3};
    printf("%d,", *a[2]);
    printf("%d,", a[2][0]);
    printf("%d ", **(a+1+('b'-'a')));
    return 0;
}
```

`ASCII value of a=97 and b=98`

Output▼

(A) 1024,1,1   (C) 1024,2,1024

(B) 1,1,1   (D) None of these

14. Select the desire output of the following code snippet with reason;

```
int *fun();                int *fun(){
int main(void){             int a=10,b=20;
  int *ptr;                 int sum=0;
  ptr=fun();                sum=sum+a+b;
  printf("%d\n",*ptr);      return &sum;
  return 0;                 }
}
```

Output with reason▼

(A) Unexpected be-   (C) 30
    havoir

(B) Address of sum   (D) None of these

15. Select the desire output of the following code snippet with reason;

```
int *fun();                int *fun(){
int main(void)             int a=10,b=20;
{                          int *sum,c;
  int *ptr=fun();          sum=&c;
  printf("%d\n",*ptr);     *sum=a+b;
  return 0;                return sum;
}                          }
```

Output with reason▼
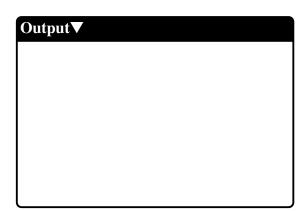
(A) Unexpected be-   (C) 30
    havoir

(B) Address of sum   (D) None of these

16. Select the output of the following program assuming size of integer is 4 bytes.

```
int main(){
 int a[3][4]={1,2,3,4,4,3,2,1,7,8,9,0};
 printf("%p %p\n", a, a+1);
  printf("%p %p\n", &a,&a+1);
 return 0;
}
```
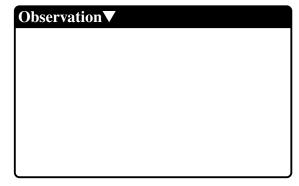
Output▼

17. Write the output of the following program.

```
int a[2][2]={1,2,3,4},i,j;
int *p[]={*a,*a+1,*a+2};
for(i=0;i<2;i++){
   for(j=0;j<2;j++){
       printf("%d ", *(*(p+i)+j));
       printf("%d ", *(*(i+p)+j));
       printf("%d ", *(*(p+j)+i));
       printf("%d\n", *(*(j+p)+i));
   }
 }
 return 0;
}
```

Output▼

18. Make a close study of the following code snippet.

Observation▼

```
int main()
{
 int b=65;
 void p=b;
 printf("%d",p);
 return 0;
}
```
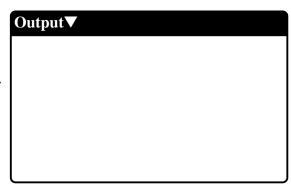
19. Select the output of the following program.

```c
int main(){
 int b=65;
 void *p=&b;
 int *j=(int *)p;
 char *ch=(char *)p;
 printf("%d  %c\n",*j,*ch);
 return 0;
}
```

Output▼

(A) 65 65

(B) 65 A

(C) Compile   time
    error

(D) Run time error

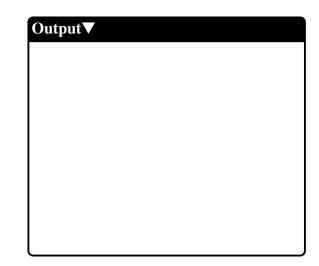20. Write the output of the code snippet.

```c
int main(){int i;
 int a[30];
 int *p=a;
 for(i=0;i<5;i++){
    scanf("%d",p+i); /* 10,20,30,40,50 */
 }
 for(i=0;i<5;i++){
  printf("%d..%d..\n",p[i],*(p+i),i[p]);
 }
 return 0;}
```
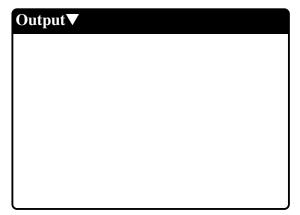
Output▼

21. Write the output of the code snippet. Assume the base address of the array is **0x7ffecb0c6490**.

```c
int main(){
 int i;
 int arr[]={11,22,33,44,55};
 int *p[]={arr, arr+1, arr+2, arr+3, arr+4};
 //is the declaration correct?
 int **ptr=p;
 for(i=0; i<5; i++){
    printf("%p %d\n",(p[0]+i),*(p[0]+i));
 }
 printf("------------------\n");
 for(int i=0;i<5;i++){
    printf("%p %d\n",((p+0)+i),*(*(p+0)+i));
 }
 return 0;
}
```

Output▼

22. Which of the given statements about the following code snippet is/are correct?

```c
int main(){int i;
 int arr[]={11,22,33,44,55};
 int *p[]={arr, arr+1, arr+2, arr+3, arr+4};
 int **ptr=p;
 int arrlen=sizeof(arr)/sizeof(arr[0]);
 for(int k=0;k<5;k++,arrlen--){
   for(int i=0;i<arrlen;i++){
    printf("%d ",*(p[k]+i));
   }
   printf("\n");
 }
 return 0;
}
```

Output▼

23. Which of the following statements are true?.

```
(1) (void *)0 is a void pointer
(2) (void *)0 is a NULL pointer
(3) int *p=(int *)0; p is a NULL pointer
(4) a[i]==i[a]
(5) a[i][j]== *(*(a+i)+j)
```

Output▼

24. Check the error or output of the following program?

```
int main(){              (i)    20 20 20
void *p;                 (ii)   20 30 20
int *i=20;               (iii) compile error
p=&i;                           at line-4
void *q=p; //line-4      (iv)  compile error
//line-5                        at line-5
printf("%d %d %d\n",i,*p,*q);
}
```

Output▼

25. Write the output of the following program? Assume that the base address of a given array **a** is 1000?

```
int main(){
int a[3][3]={4,5,6,7,8,9,1,2,3};
printf("%p %p %p\n",a[1]+2,*(a+1)+2,&a[1][2])
    ;
printf("%d %d %d\n",*(a[1]+2),*(*(a+1)+2), a
    [1][2]);
return 0;
}
```

Output▼

26. State the output of the code.

```
#include<stdio.h>        int main(){
int f(int n){              int (*p)(int)=f;
 while(--n>=0){            (*p)(8);
  printf("%d ",n-2);       return 0;
 }                       }
 return 1;
}
```

Output▼

27. Write the output of the given code snippet.

```
#include<stdio.h>        #include<stdio.h>
int main(){              void demo(){
   void demo();           printf("SS");
   void (*fun)();        }
   fun=demo;
   (*)fun();
   fun();
   return 0;
}
```

Output▼

28. Write the output of the given code snippet that uses pointer to function or function pointer.
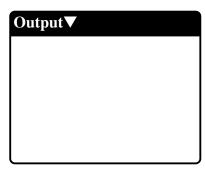
```
int fun(int x,int y){
  int z=x+y+x*y;
  return z;
}
```

```
#include<stdio.h>
int main(){
  int (*fun_ptr)(int,int);
  fun_ptr=fun;
  int x=fun_ptr(34,56);
  printf("%d\n",x);
  return 0;
}
```

Output▼

29. Mention the output of the following code snippet. [Array of pointers to function returning int type].

```
#include<stdio.h>                int fun1(int x,int y){
int main(){                        return x+y;
int x,y;                         }
int (*fun_ptr[2])(int,int);
fun_ptr[0]=fun1;                 int fun2(int x,int y){
x=fun_ptr[0](4,5);;                return x*y;
fun_ptr[1]=fun2;                 }
y=(*fun_ptr[1])(4,5);
printf("%d...%d\n",x,y);
return 0;
}
```

Output▼

30*. Find out the correct syntal(s) for making a constant pointer (i.e. The value of the pointer is constant and pointer cannot be modified).

```
(1) const <data_type> * ptr;
(2) <data_type> * const ptr;
(3) <dat_type> const *ptr;
(4) <data_type> const * const fun_ptr
(5) None of these
```

Output▼

31*. Find out the correct syntal(s) for a pointer to constant (i.e. The pointer cannot able to change the value of the variable/array that it points).

```
(1) const <data_type> * ptr;
(2) <data_type> * const ptr;
(3) <dat_type> const *ptr;
(4) <data_type> const * const fun_ptr
(5) None of these
```

Output▼

32*. Select the correct way of declaring and initializing pointer to function (i.e. function pointer).

```
(1) int (*ptr)(int,int,int)=funname;
(2) int *ptr(int,int,int)=funname;
(3) int (*ptr)(int,int,int)=&funname;
(4) (int *) ptr(int,int,int)=funname;
(5) None of these
```

Output▼
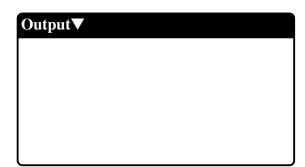
33. Find the output of the code snippet.

```
int main(){
  int a[][2][4]={5,6,7,8,9,11,12,1};
  printf("%d\n",*(*(*(a+0)+1)+2));
  return 0;
}
```

Output▼

34. Describe the output for the following code snippet.

```c
void fun(int arr[][3]){
   printf("%d\n",*(*(arr+2)+1));
   printf("%p\n",(*arr)+2);
   printf("%p\n",&arr[0][2]);
   printf("%d\n",*(((*arr)+1)+1));
}
int main(){
   int a[][3]={5,6,7,8,9,4,3,2,1};
   fun(a);
   return 0;
}
```

Output▼

35*. Explain the below declaration(s).

```c
(1) int process(int (*pf)(int a, int b)) ;
(2) int (*fun(int, void (*ptr)())) ();
(3) int *(*p)(int (*a)[ ]);
(4) int (*p)[10];
(5) float *p[20];
(6) int p(char *a);
(7) int (*p(char * a))[l0];
(8) int * (*p [10]) (char  *a);
.................
```

Output▼