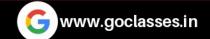# Question: 1

Consider the following code:

```c
int a[10] = { 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 };
int *b = a + 4;
int *c = &(a[4]);
```

Which of the following is true?

○       b and c both contain the integer 14

○       b and c both contain pointers to the 5th position in array a (i.e., they point to the 14)

○       b contains a pointer to the 2nd position in array a (i.e., pointing to the 11) and c contains a pointer to the 5th position in array a

○       This code will not compile

# Question : 2

Assume that the following C declarations are in effect:

```c
int a[4] = {6, 4, 1, 2};
int b[8] = {9, 8, 11, 10, 5, 7, 0, 3};
int *p = &a[1];
int *q = b;
int *r = b + 2;
```

Give the value of each of the following expressions. If an expression is illegal, give ILLEGAL as the answer. (Consider an expression to be illegal if does not adhere to recommended best practices.) If an expression is legal but has an undefined value, give UNDEFINED as the answer.

(a) *p

(b) *q

(c) *r

(d) p + q

(e) r - q

(f) *(p + 1)

(g) q[3]

(h) b - a

# Question : 3

**Problem 3: Strings and pointers (10 points)**

(a) (4pts) Consider the following code, compiled using the compiler and settings we have been using for this class.

```
char *str = "Stanford University";
char a = str[1];
char b = *(char*)((int*)str + 3);
```

What are the **char** values of variables **a** and **b** ?

Write "ERROR" across the box if the line of code declaring the variable won't compile.

a [        ]          b [        ]

## Question 4:

```c
#include <stdio.h>
int main() {

    int a[] = {1,2,3,4,5,6,7,8,9,10};

    int *ip = a+2;
    int *p1, *p2;

    p1 = (int *)((short*) ip + 4);

    p2 = (int *)((short *) (ip-2)+2);

    printf("p1 = %d\n", *p1);
    printf("p2 = %d\n", *p2);


}
```

## Question 5:

```c
#include <stdio.h>
int main() {

    int a[] = {1,2,3,4,5,6,7,8,9,10};

    int *qp = (int*)((short *)(a+5)+2);

    printf("%d", *qp);
}
```

**Question 6:**

**Problem 9: Strings and pointers**

(c) (6pts) Consider the following code, compiled using the compiler and settings we have been using for this class.

```
char str [ ] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

char a = str[0];
char c = *(char *)((int *)str + 2);
char d = (str + 2)[3];
```

Note: recall that type short is like int but 2 bytes.

What are the char values of variables a, b, c, and d? (a is filled in for you as an example) Write "ERROR" across the box if the line of code declaring the variable won't compile, or executing the operation could give a memory error (e.g., off end of array).

a    'A'                    c   [ ]

d    [ ]

# Question 7:

Consider the following C declaration:

```
int iarr[100];
int *p = iarr;
```

Which of the following expressions is semantically equivalent to "iarr[50]"?

(a) *(int *)((char *)p + 50 * sizeof(int))

(b) *(int *)(p + 50 * sizeof(int *))

(c) ((int *)((char *)p + 50))[0]

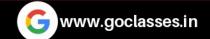(d) *(char *)((int *)p + 50)

# Question 8:

**Question 6.** [10 points] What output is printed by the following C program? Assume that `sizeof(int) == 4`. Explain briefly.

```c
int a[4] = { 6, 7, 8, 9 };
printf("%d\n", (int) (&a[2] - &a[0]));
printf("%d\n", (int) ((char *)&a[2] - (char *)&a[0]));
```

https://jhucsf.github.io/spring2023/resources/exam01-fall2021.pdf

# Question 9:

```c
#include <stdio.h>

int main() {

    char *str = (char*)1000;
    long *u_ptr = (long*)str;
    int *i_ptr = (int *)(u_ptr + 1);
    char *c_ptr = (char *)(i_ptr + 2);

    printf("u_ptr = %p\n", u_ptr);
    printf("i_ptr = %p\n", i_ptr);
    printf("c_ptr = %p\n", c_ptr);

}
```

## Question 10:

```c
#include<stdio.h>
int main()
{
    int x[] = {12, -1, 3, 8, -4};
    long *y = (long *)x;

    int *z = (int *) ((char *)(y+2)-sizeof(int));

    printf("%d", z[-1]);

}
```