Practice questions from Recursion and Pointers(basic):

Question1:
Consider the following code snippet that is the base to **find and show the number of times the letter 'd' appears in the string. You are required to write a recursive function that matches the prototype in the program. Also provide dry-run and function stack for the given string.**

```
#include <iostream>

using namespace std;

// Function prototype

int numChars(char, char [], int);

int main()

{

  char array[] = "abdcdedf";


  /* Display the number of times the letter 'd' appears in the string. */

  cout << "The letter d appears "

  << numChars('d', array, 0) << " times.\n";

  return 0;}
```

Question 2:

One of the most famous mathematical recursive algorithms is the Fibonacci sequence. Fibonacci sequences appear in many places in nature, such as branching of trees, the spiral of shells, the fruitlets of a pineapple, an uncurling fern frond, and the arrangement of a pine cone. You are given a program that computes the nth element in Fibonacci sequence. Provide a complete dry run and function stack for the program. Assume the user enters n=5.

```
#include <iostream>
using namespace std;

int fib(int n) {
     if (n <= 0)         // base case
       return 1;
     else if (n==1)    // base case
       return 1;
     else
       return fib(n-1) + fib(n-2);
}

int main() {
    int n;
    cin>>n;
    cout<<n<<"th Fibonacci number is: "<<fib(n);
    return 0;
}
```

Question 3:

Provide output of the following code snippets. Assume the code is error free.

```
int main()
{
  int num[5]={1,2,3,4,5};
  int *p ;
  p = num;
  *p = 20;
  p = &num[1];
  *(++p) = 30;
  p = num + 4;
  *p = 30;
  p = num;
  *(p + 3) = 40;
  for(int i = 1; i< 5; i++)
    cout<< num[i]<<"  ";
  return 0;
}
```

```
   #include <iostream>
char c[7][11]= {"PF-final", "students", "happy", "2024",
"program", "semester", "end"};
char* add(char* ptr){
    return ptr + 11;
}
char* sub (char* ptr){
    return ptr - 11;
}
using namespace std;
int main()
{
    char* mystery = c[4];
    cout<< mystery<<endl;
    cout<< sub(mystery)[2]<<endl;
    mystery = sub(mystery);
    cout<<mystery<<endl;
    cout<<sub(mystery)+ 1<<endl;
    cout<<add(add(mystery))+ 13<<endl;
    cout<<*add(add(mystery))<< endl;
    return 0;
}
```

```
#include <iostream>
 using namespace std;
 int main()
 {
   int arr[] = {4, 5, 6, 7};
   int *p = (arr + 1);
   cout << *p;
   return 0;
    }
```

```
#include <iostream>
   using namespace std;
   int main ()
   {
       int numbers[5];
       int * p;
       p = numbers;  *p = 10;
       p++;  *p = 20;
       p = &numbers[2];  *p = 30;
       p = numbers + 3;  *p = 40;
       p = numbers;  *(p + 4) = 50;
       for (int n = 0; n < 5; n++)
           cout << numbers[n] << ",";
       return 0;
   }
```