1. (20 points) After each segment of code executes, give a list of all variables defined and their corresponding values. For pointer variables, whose address do they contain (i.e. where do they point – draw a memory diagram if that is easier than saying in words)?

a) char s[80], t[] = “yo”, *p;
   int i;
   sprintf(s, “%s%s%3d”, t, t, 81);
   for(p=s; *p != ‘\0’; p++){}

b) double x[] = {1.0, 2.0, 3.0, 4.0, 5.0};
   int i;
   for(i=0; i<5; i+=2)
   {
      x[i] = pow(x[i], 5-i);
   }

c) char s[] = “Hawaii”, ch1, ch2, ch3, *p, *q;
   p = s + 1;
   ch1 = *p;
   ch2 = *(++p);
   ch3 = *(p + 1);
   *q = *(s + 6);
2. (24 points) Print the output from the following segments of C-code (exactly in the order it will appear when run):

a) int i=1, j=10, count=0;
   while(1)
   {
   if(i >= j) break;
   else if(count++%2)
      printf("i = %d, j = %d\n", i, j)
   
   i *= 2;
   j += 1;
   }

b) int i, *x, *q;
   char s[] = "decaf", *p
   x = (int *)malloc(5*sizeof(int));
   for(p = s, q=x; *p != 0; p++, q++)
   {
   *q = *p – 'a';
   }
   for(i=0; i<5; i++)
   {
   printf("x[%d] = %d\n", i, x[i]);
   }
   free(x);

c) int i, count, x[] = {1, 2, 3, 4, 5};
   
   for(i=0, count=0; count<5; count++, i = (i+3)%5)
   {
   x[i] = x[i] + count;
   printf("x[%d] = %d, count = %d\n", i, x[i], count);
   }
3. (24 points) Fix the “bugs” in the following program (rewrite the program below). The intention of
the program is to prompt the user to enter a string and print whether the string is palindrome or not. A
string is palindrome if it is identical to itself when its characters are taken in reverse order. Examples
of palindrome strings are: “radar”, “racecar”, “abba”.

```c
void isPalindrome(char s[])
{
    int N = strlen(s), *p, *q;

    for(p=s, q=s+N; p>q; p++, q--)
    {
        if(&p == &q) return 1;
        else return 0;
    }
}

int main()
{
    char s[];

    printf(“Enter a string >\n”);
    scanf(“%s”, &s);

    if(isPalindrome(s)) printf(“%d is palindrome.\n”, s);
    else printf(“%d is not palindrome.\n”, s);

    return 0;
}
```
4. (32 points) Write a program that prompts the user to enter two strings, \texttt{s1,} and \texttt{s2}. The strings are then compared to find characters that are common to both, and the result is printed to the screen. Be careful not to print duplicate characters. It would be helpful to create a helper function, \textbf{int isCharFound(char s[], char ch)}, that returns 1 if \texttt{ch} is found in string, \texttt{s}, otherwise returns 0 if \texttt{ch} not found in \texttt{s}. You may also use any build-in functions in \texttt{<string.h>}. See sample input and output below:

// example 1
Enter two strings >
howdy  ho
Common characters:
ho

// example 2
Enter two strings >
zebra  baseball
Common characters:
eba