1. DAY 3 II PROGRAM

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
   char str1[20], str2[20];
   int len, len1, len2, i, j, found=0, not_found=0;
   printf("Enter first string: ");
   gets(str1);
   printf("Enter second string: ");
   gets(str2);
   len1 = strlen(str1);
   len2 = strlen(str2);
   if(len1 == len2)
   {
      len = len1;
      for(i=0; i<len; i++)
      {
         found = 0;
         for(j=0; j<len; j++)
         {
            if(str1[i] == str2[j])
            {
               found = 1;
               break;
            }
         }
         if(found == 0)
         {
            not_found = 1;
            break;
         }
```

```
    }
    if(not_found == 1)
        printf("\nStrings are not Anagram");
    else
        printf("\nStrings are Anagram");
  }
  else
    printf("\nBoth string must contain same number of character to be an Anagram Strings");
  getch();
  return 0;
}
```

2.

```
#include <stdio.h>
#include <string.h>
#define MAX_STRING_LENGTH 100
void longestCommonPrefix(char strs[][MAX_STRING_LENGTH], int strsSize, char
prefix[MAX_STRING_LENGTH]) {
  if (strsSize == 0) {
    strcpy(prefix, "");
    return;
  }
       int i,j;
  int len = strlen(strs[0]);

  for ( i = 0; i < len; i++) {
    for ( j = 1; j < strsSize; j++) {
      if (strs[j][i] != strs[0][i]) {
        prefix[i] = '\0';
        return;
      }
    }
    prefix[i] = strs[0][i];
  }
```

```
    prefix[len] = '\0';
}
int main() {
    int strsSize;
            int i;
    printf("Enter the number of strings: ");
    scanf("%d", &strsSize);
    char strs[strsSize][MAX_STRING_LENGTH];
    char prefix[MAX_STRING_LENGTH];
    for ( i = 0; i < strsSize; i++) {
        printf("Enter string %d: ", i + 1);
        scanf("%s", strs[i]);
    }
   longestCommonPrefix(strs, strsSize, prefix);
    printf("Longest common prefix: %s\n", prefix);
   return 0;
}
```

3.

```
#include <stdio.h>
#include <string.h>
const char phoneMap[10][5] = {"", "", "abc", "def", "ghi", "jkl", "mno", "pqrs", "tuv", "wxyz"};
void backtrack(char *digits, int index, char *combination, char result[][5], int *count) {
    if (digits[index] == '\0') {
        combination[index] = '\0';
        strcpy(result[(*count)], combination);
        (*count)++;
        return;
    }
  int digit = digits[index] - '0';
    char *letters = phoneMap[digit];
            int i;
```

```
    for ( i = 0; i < strlen(letters); i++) {
        combination[index] = letters[i];
        backtrack(digits, index + 1, combination, result, count);
    }
}
void letterCombinations(char *digits) {
    int n = strlen(digits);
    if (n == 0) {
        printf("No digits entered.\n");
        return;
    }
            int i,j;
    int size = 1;
    for ( i = 0; i < n; i++) {
        size *= strlen(phoneMap[digits[i] - '0']);
    }
    char result[size][5];
    int count = 0;
    char combination[5]; // assuming the maximum length of a combination is 4
    backtrack(digits, 0, combination, result, &count);
     printf("Letter combinations:\n");
    for ( j = 0; j < count; j++) {
        printf("%s ", result[j]);
    }
    printf("\n");
}
int main() {
    char digits[100];
   printf("Enter the digits: ");
    scanf("%s", digits);
 letterCombinations(digits);
 return 0;
}
```