1)Write a function with macro that takes argument dynamically

Sol:

```
#include <stdio.h>
#define print(...) printf(__VA_ARGS__)

int main()
{
   char a[10],b[10],c[10];
   printf("Enter the string1:");
   scanf("%[^\n]s",a);
   printf("Enter the string2:");
   getchar();
   scanf("%[^\n]s",b);
   printf("Enter the string3:");
   getchar();
   scanf("%[^\n]s",c);

   print("Welcome to %s\n",a);
   print("Hello to %s:Welcome to %s\n",b,c);

   return 0;
}
```

2)Write a program to replace the given value with the value above it in a matrix using c:

Sol:

```
#include<stdio.h>
int main()
{
   int r,m;
   printf("Enter the dimension of the matrix:");
   scanf("%d",&r);
   int a[r][r];
   for(int i=0;i<r;i++)
   {
      for(int j=0;j<r;j++)
      {
         printf("The element %d%d:",i,j);
         scanf("%d",&a[i][j]);

      }

   }
   printf("The array before changes>>");
   printf("\n");
   for(int i=0;i<r;i++)
   {
      for(int j=0;j<r;j++)
      {
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }
    printf("Enter the element to replace:");
    scanf("%d",&m);
     for(int i=1;i<r;i++)
    {
        for(int j=0;j<r;j++)
        {
            if(m==a[i][j])
            {
                a[i][j]=a[i-1][j];
            }



        }
    }
    printf("The array after changes>>");
    printf("\n");
     for(int i=0;i<r;i++)
    {
        for(int j=0;j<r;j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }


    return 0;
}
```

3) Write a program to sort the given array

Sol:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#define MAX_LEN 100
char* sort_string(char* str)
{
   int len = strlen(str);
    char** substrs = malloc(len * sizeof(char*));
    int n_subsets = 0;
    for (int i = 0; i < len; )
    {
        int j = i;
        while (j < len && isdigit(str[j]) == isdigit(str[i]))
        {
            j++;
        }
        substrs[n_subsets] = malloc((j - i + 1) * sizeof(char));
        strncpy(substrs[n_subsets], &str[i], j - i);
        substrs[n_subsets][j - i] = '\0';
        i = j;
        n_subsets++;
    }

    for (int i = 0; i < n_subsets; i++)
    {
        int len = strlen(substrs[i]);
        for (int j = 0; j < len; j++)
        {
            for (int k = j + 1; k < len; k++)
            {
                if (substrs[i][j] > substrs[i][k])
                {
                    char temp = substrs[i][j];
                    substrs[i][j] = substrs[i][k];
                    substrs[i][k] = temp;
                }
            }
        }
    }

    char* sorted_str = malloc((len + 1) * sizeof(char));
    int index = 0;
    for (int i = 0; i < n_subsets; i++)
    {
        int len = strlen(substrs[i]);
        strncpy(&sorted_str[index], substrs[i], len);
        index += len;
        free(substrs[i]);
    }
    sorted_str[len] = '\0';
    free(substrs);
    return sorted_str;
}

int main()
{
    char str[MAX_LEN];
    printf("Enter a string: ");
    fgets(str, MAX_LEN, stdin);
    str[strcspn(str, "\n")] = '\0';
    char* sorted_str = sort_string(str);
    printf("Sorted string: %s\n", sorted_str);
    free(sorted_str);
    return 0;
}
```