1. Develop a Program in C for the following:

a) Declare a calendar as an array of 7 elements (A dynamically Created array) to represent 7 days of a week. Each Element of the array is a structure having three fields. The first field is the name of the Day (A dynamically allocated String), The second field is the date of the Day(A integer), the third field is the description of the activity for a particular day (A dynamically allocated String).

b) Write functions create(), read() and display(); to create the calendar, to read the data fromthe keyboard and to print weeks activity details report on screen.

NOTE:

1. A Dynamic Array is allocated memory at runtime and its size can be changed later in the program.
2. We can create a dynamic array in C by using the following methods: Using malloc() Function.
   Using calloc() Function.
   Resizing Array Using realloc() Function.

3. **1. Dynamic Array Using malloc() Function**
   The **"malloc"** or **"memory allocation"** method in C is used to dynamically allocate a single large block of memory with the specified size. It returns a pointer of type void which can be cast into a pointer of any form. It is defined inside **<stdlib.h>** header file.

Syntax:
ptr = (cast-type*) malloc(byte-size);

Example:
ptr = (int*) malloc(100 * sizeof(int));

2. Dynamic Array Using calloc() Function
The **"calloc"** or **"contiguous allocation"** method in C is used to dynamically allocate the specified number of blocks of memory of the specified type and initialized each block with a default value of 0.

The process of creating a dynamic array using calloc() is similar to the malloc() method. The difference is that calloc() takes arguments instead of one as compared to malloc(). Here, we provide the size of each element and the number of elements required in the dynamic array. Also, each element in the array is initialized to zero.

Syntax:
ptr = (cast-type*)calloc(n, element-size);

Example:
ptr = (int*) calloc(5, sizeof(float));

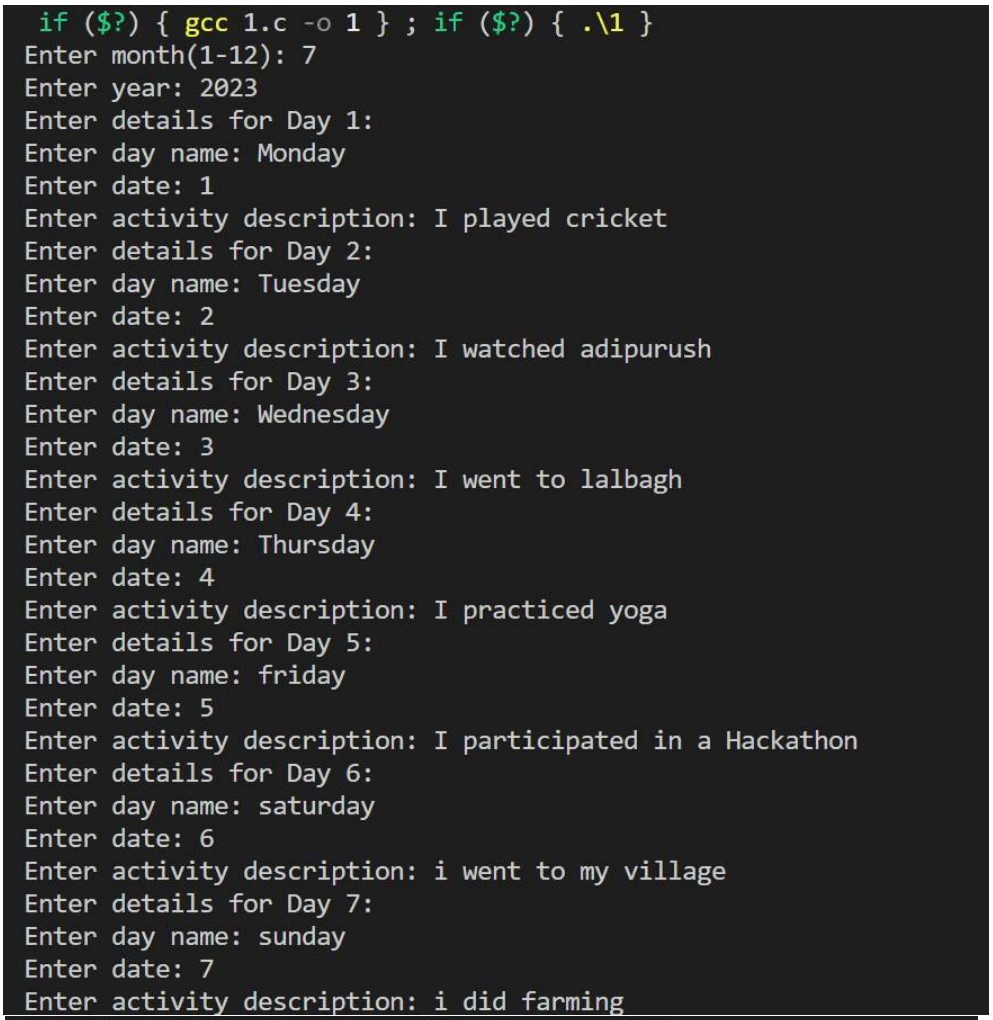
3. Dynamically Resizing Array Using realloc() Function

The **“realloc”** or **“re-allocation”** method in C is used to dynamically change the memory allocation of a previously allocated memory.
Using this function we can create a new array or change the size of an already existing array.

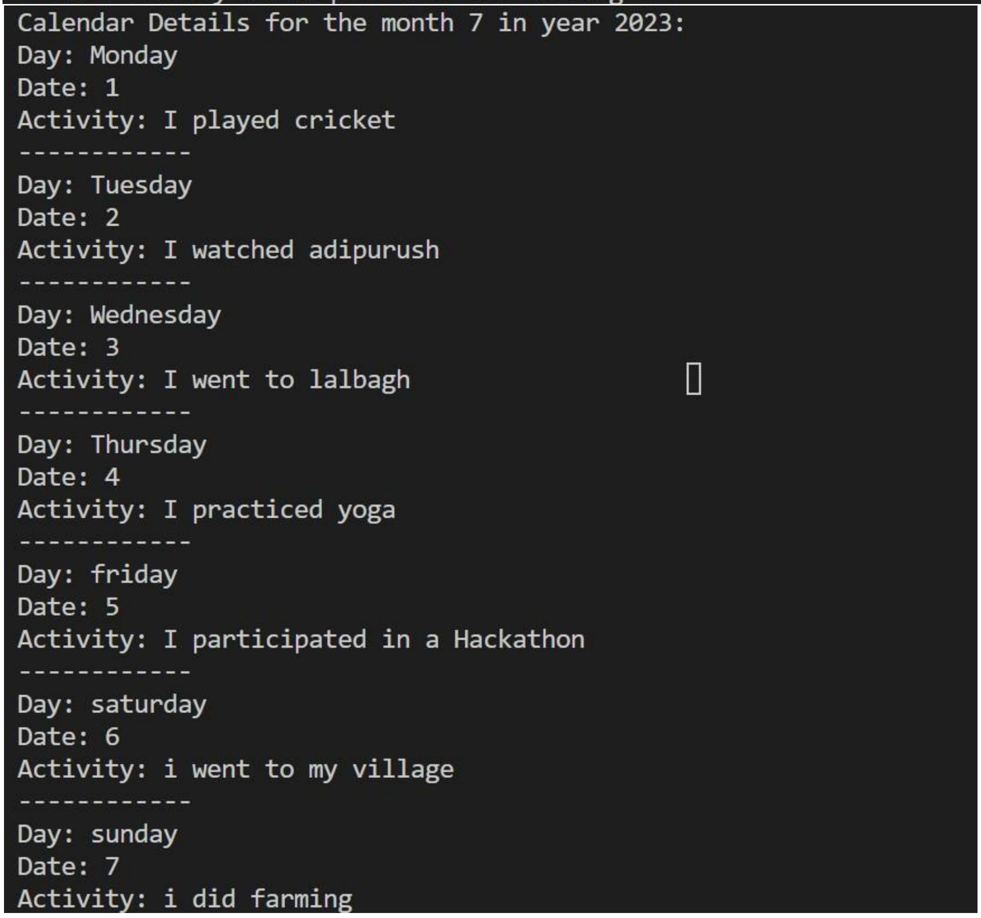
Syntax:
ptr = realloc(ptr, newSize);

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct Day
 {
char *dayName;
 int date;
char *activityDescription;
};

struct Day* create()
 {
struct Day *newDay = malloc(sizeof(struct Day));
 if (newDay == NULL)
{
printf("Memory allocation failed.\n");
exit(EXIT_FAILURE);
}
newDay->dayName = malloc(20);
 if (newDay->dayName == NULL)
 {
printf("Memory allocation failed.\n");
 exit(EXIT_FAILURE);
}
newDay->activityDescription = malloc(100);
if (newDay->activityDescription == NULL)
 {
printf("Memory allocation failed.\n");
exit(EXIT_FAILURE);
}
return newDay;
}


void read(struct Day *day)
{
 printf("Enter day name: ");
scanf("%s", day->dayName);
printf("Enter date: ");
scanf("%d", &day->date);
while (getchar() != '\n');
 printf("Enter activity description: ");
fgets(day->activityDescription, 100, stdin);
day->activityDescription[strcspn(day->activityDescription, "\n")] = '\0';
// “complementary span”
}

void display(const struct Day *day)
{
printf("Day: %s\n Date: %d\n Activity: %s\n", day->dayName,
day->date, day->activityDescription);
 printf("  \n");
}

void freeDay(struct Day *day)
 {
free(day->dayName);
free(day->activityDescription);
 free(day);
}

int main()
 {
const int daysPerWeek = 7;
struct Day *myCalendar[daysPerWeek];
 int month, year;
printf("Enter month(1-12): ");
 scanf("%d", &month);
printf("Enter year: ");
scanf("%d",&year);
for (int i = 0; i < daysPerWeek; ++i)
 {
myCalendar[i] = create();
printf("Enter details for Day %d:\n", i + 1);
 read(myCalendar[i]);
}
```

```
printf("\nCalendar Details for the month %d in year %d:\n",month,year);
for (int i = 0; i < daysPerWeek; ++i)
{
display(myCalendar[i]);
freeDay(myCalendar[i]);
}
return 0;
}
```



```
 if ($?) { gcc 1.c -o 1 } ; if ($?) { .\1 }
Enter month(1-12): 7
Enter year: 2023
Enter details for Day 1:
Enter day name: Monday
Enter date: 1
Enter activity description: I played cricket
Enter details for Day 2:
Enter day name: Tuesday
Enter date: 2
Enter activity description: I watched adipurush
Enter details for Day 3:
Enter day name: Wednesday
Enter date: 3
Enter activity description: I went to lalbagh
Enter details for Day 4:
Enter day name: Thursday
Enter date: 4
Enter activity description: I practiced yoga
Enter details for Day 5:
Enter day name: friday
Enter date: 5
Enter activity description: I participated in a Hackathon
Enter details for Day 6:
Enter day name: saturday
Enter date: 6
Enter activity description: i went to my village
Enter details for Day 7:
Enter day name: sunday
Enter date: 7
Enter activity description: i did farming
```



```
Calendar Details for the month 7 in year 2023:
Day: Monday
Date: 1
Activity: I played cricket
------------
Day: Tuesday
Date: 2
Activity: I watched adipurush
------------
Day: Wednesday
Date: 3
Activity: I went to lalbagh
------------
Day: Thursday
Date: 4
Activity: I practiced yoga
------------
Day: friday
Date: 5
Activity: I participated in a Hackathon
------------
Day: saturday
Date: 6
Activity: i went to my village
------------
Day: sunday
Date: 7
Activity: i did farming
```

2.Develop a Program in C for the following operations on Strings.

a. Read a main String (STR), a Pattern String (PAT) and a Replace String (REP)

b. Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report suitable messages in case PAT does not exist in STR

Support the program with functions for each of the above operations. Don't use Built-infunctions.

```
#include<stdio.h>
#include<string.h>

void findandreplace(char[],char[],char[]);

main()
{
      char str[100],pat[100],rep[100];
    printf("\nEnter Main string \n");
      gets(str);
      printf("\nEnter the string to be searched \n");
      gets(pat);
      printf("\nEnter the string to be replaced \n");
      gets(rep);
  findandreplace(str,pat,rep);
}
```

```
void findandreplace(char str[100],char pat[100],char rep[100])
{
      int i,j,c,m,k,found=0;
      char res[100];
      i=m=c=j=0;
      while(str[c]!='\0')
      {
            if(str[m]==pat[i])    // Character Matched
            {
                  i++; m++;
                  if(pat[i]=='\0') //Pattern Found
                  {
                        found=1;
                        printf("Pattern found at %d\n",c);
                        for(k=0;rep[k]!='\0';k++,j++)
                        // Copy Replace String in result String
                        {
                              res[j]=rep[k];
                        }
                              i=0;
                        c=m;
                  } }
            else  //Pattern Not Found
            {
                  res[j] = str[c];
                  j++; c++;
                  m=c;
                  i=0;
            }
      } // end while
      res[j]='\0';
      if(found)
      {
      printf("\nThe resultant string is:\n%s\n" ,res);
      }
      else
            printf("Pattern not found\n");
}
```

Output:

```
1.Enter any string
im at village now and i will come back from village by tomorrow
Enter the string to be searched
village
Enter the string to be replaced
```

```
town
Pattern found at 6
Pattern found at 44

The resultant string is:
im at town now and i will come back from town by tomorrow



2. Enter any string
glass
Enter the string to be searched
glass
Enter the string to be replaced
class
Pattern found at 0

The resultant string is:
Class


3. Enter any string
apple
Enter the string to be searched
p
Enter the string to be replaced
b
Pattern found at 1
Pattern found at 2

The resultant string is:
abble


4. Enter Main string
im at town
Enter the string to be searched
to
Enter the string to be replaced
bo
Pattern found at 6
The resultant string is:
im at bown

5.Enter Main string
hello naveen
Enter the string to be searched
naveen
Enter the string to be replaced
```

```
navin
Pattern found at 6
The resultant string is:
hello navin

6.Enter any string
hi
Enter the string to be searched
a
Enter the string to be replaced
b
Pattern not found
```

6.Develop a menu driven Program in C for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX)

a. Insert an Element on to Circular QUEUE

b. Delete an Element from Circular QUEUE

c. Demonstrate Overflow and Underflow situations on Circular QUEUE

d. Display the status of Circular QUEUE

e. Exit

Support the program with appropriate functions for each of the above operations.

```c
#include<stdio.h>
#include<stdlib.h>
#define MAX 5

void insert(int);
int delet(void);
void display(void);
int a[MAX],f=0,r=-1,count=0;

void main()
{
    int ch,item,itemdel;
     printf("Queue Operations Demo");
    while(1)
    {
        printf("\nEnter Your Choice :");
        printf("1.Insert\t2.Delete\t3.Display\t4.Exit\n");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1: printf("Enter the item\n");
                scanf("%d",&item);
                insert(item);
                break;
            case 2: itemdel=delet();
                if(itemdel)

printf("The deleted item is %d\n",itemdel);
                else
                    printf("Queue underflow\n");
                break;
            case 3: display();
                break;
            case 4: exit(0);
        }
    }
}

void insert (int item)
{
    if(count==MAX)
        printf("Q Overflow\n");
    else
    {
r=(r+1)%MAX;
        a[r]=item;
        count=count+1;
    }
}

int delet(void)
{
    int itemdel;
    if(count==0)
        return 0;
    else
    {
        itemdel=a[f];
        f=(f+1)%MAX;
        count=count-1;

return (itemdel);
    }
}
```

```
void display(void)
{
    int i,j;
    if(count==0)
        printf("Q empty\n");
    else
    {
        i=f;
        for(j=1;j<=count;j++)
        {
            printf("%d\t",a[i]);
            i=(i+1)%MAX;
        }
    }
}
```

```
Queue Operations Demo
Enter Your Choice :1.Insert 2.Delete   3.Display  4.Exit
2
Queue underflow

Enter Your Choice :1.Insert 2.Delete   3.Display  4.Exit
3
Q empty

Enter Your Choice :1.Insert 2.Delete   3.Display  4.Exit
1
Enter the item
10
Enter Your Choice :1.Insert 2.Delete   3.Display  4.Exit
1
Enter the item
25
Enter Your Choice :1.Insert 2.Delete   3.Display  4.Exit
1
Enter the item
40
Enter Your Choice :1.Insert 2.Delete   3.Display  4.Exit
3
10    25    40
Enter Your Choice :1.Insert 2.Delete   3.Display  4.Exit
2
The deleted item is 10

Enter Your Choice :1.Insert 2.Delete   3.Display  4.Exit
```

```
3
25    40
Enter Your Choice :1.Insert 2.Delete    3.Display  4.Exit
```