```
/*
* Que.1 : Sum of elements of array using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        int *p;
        int n , i;
        int sum = 0; //to stores the sum of elements

        printf("enter how many elements you want to insert : ");
        scanf("%d",&n);

        p = (int*)malloc(n*sizeof(int));

        printf("enter array elements\n");
        for(i = 0 ; i < n ; i++)
        {
                scanf("%d",p+i);
        }

        printf("array elements are\n");
        for(i = 0 ; i < n ; i++)
        {
                sum = sum + *(p+i); //adding elements in sum
                printf("%d ",*(p+i));
        }

        printf("\nsum of array elements is : %d",sum);
        getch();
}
```

```c
/*
* Que.2 : Accepet string from user and print as it is using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char ch;
        int no_of_char = 1;
        int i = 0;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter a string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');

        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }

        getch();
}
```

```
/*
* Que.3 : Accepet string from user with multiple spaces and print single space as
delimeter using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char ch;
        int no_of_char = 1;
        int i = 0;
        int cnt = 0;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter a string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');

        printf("your string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }

        printf("\nexpected string is : \n");
        i = 0;
        while(*(str+i) != '\0')
        {
                while(*(str+i) == ' ')
                {
                        i++;
                }
                if(*(str+i) != ' ' && *(str+i) != '\0')
                {
                        cnt++;
                        if(cnt == 1);
                        else
                                printf(" ");
        }
        while(*(str+i) != ' ' && *(str+i) != '\0')
        {
            printf("%c",*(str+i));
            i++;
        }
    }

    getch();
}
```

```
/*
* Que.4 : Return position of palindrome elements of array using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        int *p;
        int n , i;
        int temp; //temporary veriable
        int rvs; // to stored the reverse of element
        int rmd; //to stored the remainder

        printf("enter how many elements you want to insert : ");
        scanf("%d",&n);

        p = (int*)malloc(n*sizeof(int));

        printf("enter array elements\n");
        for(i = 0 ; i < n ; i++)
        {
                scanf("%d",p+i);
        }

        for(i = 0 ; i < n ; i++)
        {
                temp = *(p+i);
                //here we reverse the element
                rvs = 0;
                while(temp != 0)
                {
                        rmd = temp % 10;
                        rvs = rvs * 10 + rmd;
                        temp = temp/10;
                }
                // here we check given element is palindrome if it is palindrome print its
index
                if(*(p+i) == rvs)
                {
                        printf("\npallindrome element %d position at %d index",*(p+i),i);
                }

        }

        getch();
}
```

```
/*
* Que.5 : Print 1st half ascending elements and 2nd half descending elements of array
using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       int *p;
       int n , i , j ;
       int m; //middle index of array
       int temp; //temporary variable

       printf("enter how many elements you want to insert : ");
       scanf("%d",&n);

       p = (int*)malloc(n*sizeof(int));

       printf("enter array elements\n");
       for(i = 0 ; i < n ; i++)
       {
              scanf("%d",p+i);
       }

       printf("\n1st half ascending elements and 2nd half descending elements of array :
\n");

       m = n/2; // middle index

       // here we sort 1st half array asceding
       for(i = 0 ; i <= m ; i++)
       {
              for(j = i + 1 ; j <= m ; j++)
              {
                     if(*(p+i) > *(p+j))
                     {
                         temp = *(p+i);
                         *(p+i) = *(p+j);
                         *(p+j) = temp;
                  }
              }
       }

       // here we sort 2nd half array desceding
       for(i = m+1 ; i < n ; i++)
       {
              for(j=i+1 ; j < n ; j++)
              {
                     if(*(p+i) < *(p+j))
                     {
                            temp = *(p+i);
                         *(p+i) = *(p+j);
                         *(p+j) = temp;
                  }
              }
       }

       for(i = 0 ; i < n ; i++)
       {
              printf("%d ",*(p+i));
       }

       getch();
}
```

```
/*
* Que.6 : Accepet string from user and print number of characters using dynamic memory
allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char ch;
        int no_of_char = 1;
        int i = 0;
        int cnt = 0;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter a string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');

        printf("your string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }

        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                if(*(str+i) >= 'a' && *(str+i) <= 'z' || *(str+i) >= 'A' && *(str+i) <=
'Z')
                {
                        cnt++;
                }
        }
        printf("\nnumber of characters in given string are %d",cnt);

        getch();
}
```

```c
/*
* Que.7 : Accept string from user and print it reverse using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       char *str = NULL;
       char ch;
       int no_of_char = 1;
       int i = 0;
       int j;

       str = (char*)malloc(sizeof(char));
       *(str+0) = '\0';

       printf("please enter a string : ");
       do{
              scanf("%c",&ch);
              if(ch != '\n')
              {
                     no_of_char++;
                     str = (char*)realloc(str,no_of_char*sizeof(char));
                     *(str+i) = ch;
                     *(str+i+1) = '\0';
                     i++;
              }
       }while(ch != '\n');
       j = i;

       printf("your string is : \n");
       for(i = 0 ; *(str+i) != '\0' ; i++)
       {
              printf("%c",*(str+i));
       }

       printf("\nreverse string is : \n");
       for(i = j ; i >= 0 ; i--)
       {
              printf("%c",*(str+i));
       }

       getch();
}
```

```c
/*
* Que.8 : Copy elements of array in to another array using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       int *p;
       int *q;
       int n , i;

       printf("enter how many elements you want to insert : ");
       scanf("%d",&n);

       p = (int*)malloc(n*sizeof(int));

       q = (int*)malloc(n*sizeof(int));

       printf("enter array elements\n");
       for(i = 0 ; i < n ; i++)
       {
              scanf("%d",p+i);
       }

       // copy elements of array into another array
       printf("\ncopied elements of array : \n");
       for(i = 0 ; i < n ; i++)
       {
              *(q+i) = *(p+i);
              printf("%d ",*(q+i));
       }

       getch();
}
```

```
/*
* Que.9 : Accepet string from user and print number of vowels and consonants using
dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       char *str = NULL;
       char ch;
       int no_of_char = 1;
       int i = 0;
       int cntv = 0;
       int cnt = 0;
       int cntc = 0;

       str = (char*)malloc(sizeof(char));
       *(str+0) = '\0';

       printf("please enter a string : ");
       do{
              scanf("%c",&ch);
              if(ch != '\n')
              {
                     no_of_char++;
                     str = (char*)realloc(str,no_of_char*sizeof(char));
                     *(str+i) = ch;
                     *(str+i+1) = '\0';
                     i++;
              }
       }while(ch != '\n');


       printf("your string is : \n");
       for(i = 0 ; *(str+i) != '\0' ; i++)
       {
              printf("%c",*(str+i));
       }

       for(i = 0 ; *(str+i) != '\0' ; i++)
       {
              if(*(str+i) >= 'a' && *(str+i) <= 'z' || *(str+i) >= 'A' && *(str+i) >=
'Z')
              {
                     if(*(str+i) == 'a' || *(str+i) == 'e' || *(str+i) == 'i' || *(str+i)
== 'o' || *(str+i) == 'u' || *(str+i) == 'A' || *(str+i) == 'I' || *(str+i) == 'O' ||
*(str+i) == 'U')
                     {
                            cntv++;
                     }
```

```
                }
        }

        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                if(*(str+i) >= 'a' && *(str+i) <= 'z' || *(str+i) >= 'A' && *(str+i) <=
'Z')
                {
                        cnt++;
                }
        }

        cntc = cnt - cntv;

        printf("\nnumber of vowels in given string are %d\n",cntv);
        printf("number of consonnats in given string are %d",cntc);

        getch();
}
```

```
/*
* Que.10: Accept string from user and reverse the string like mirror img using dynamic
memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       char *str = NULL;
       char ch;
       int no_of_char = 1;
       int i = 0;
       int temp , temp2;

       str = (char*)malloc(sizeof(char));
       *(str+0) = '\0';

       printf("please enter a string : ");
       do{
              scanf("%c",&ch);
              if(ch != '\n')
              {
                     no_of_char++;
                     str = (char*)realloc(str,no_of_char*sizeof(char));
                     *(str+i) = ch;
                     *(str+i+1) = '\0';
                     i++;
              }
       }while(ch != '\n');


       printf("your string is : \n");
       for(i = 0 ; *(str+i) != '\0' ; i++)
       {
              printf("%c",*(str+i));
       }

       printf("\nreverse string is : \n");
       i = 0;
       while(*(str+i) != '\0')
       {
              while(*(str+i) == ' ')
              {
                     printf(" ");
                     i++;
              }
              temp = i;

              while(*(str+i) != ' ' && *(str+i) != '\0')
              {
```

```
                        i++;
                }
                temp2 = i-1;

                while(temp2 >= temp && *(str+temp2) != '\0')
                {
                        printf("%c",*(str+temp2));
                        temp2--;
                }
        }
        getch();
}
```

```c
/*
* Que.11: Sort even numbers in given array using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       int *p;
       int n , i , j ;
       int temp;

       printf("enter how many elements you want to insert : ");
       scanf("%d",&n);

       p = (int*)malloc(n*sizeof(int));

       printf("enter array elements\n");
       for(i = 0 ; i < n ; i++)
       {
              scanf("%d",p+i);
       }

       printf("\neven sorted elements of array : \n");
       for(i = 0 ; i < n ; i++)
       {
              for(j = i + 1 ; j < n ; j++)
              {
                     // here we check elements are even then sort
                     if(*(p+i) % 2 == 0 && *(p+j) % 2 == 0 && *(p+i) > *(p+j))
                     {
                            temp = *(p+i);
                         *(p+i) = *(p+j);
                         *(p+j) = temp;
                     }
              }

       }

       for(i = 0 ; i < n ; i++)
       {
              printf("%d ",*(p+i));
       }

       getch();
}
```

```c
/*
* Que.12: Accept string from user and replace space with $ using dynamic memory
allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       char *str = NULL;
       char ch;
       int no_of_char = 1;
       int i = 0;

       str = (char*)malloc(sizeof(char));
       *(str+0) = '\0';

       printf("please enter a string : ");
       do{
              scanf("%c",&ch);
              if(ch != '\n')
              {
                     no_of_char++;
                     str = (char*)realloc(str,no_of_char*sizeof(char));
                     *(str+i) = ch;
                     *(str+i+1) = '\0';
                     i++;
              }
       }while(ch != '\n');


       printf("your string is : \n");
       for(i = 0 ; *(str+i) != '\0' ; i++)
       {
              printf("%c",*(str+i));
       }

       printf("\nexpected string is : \n");
       for(i = 0 ; *(str+i) != '\0' ; i++)
       {
              if(*(str+i) == ' ')
              {
                     *(str+i) = '$';
              }
              printf("%c",*(str+i));
       }

       getch();
}
```

```c
/*
* Que.13: Seperate odd and even numbers in given array using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        int *p;
        int n , i , j ;
        int temp;

        printf("enter how many elements you want to insert : ");
        scanf("%d",&n);

        p = (int*)malloc(n*sizeof(int));

        printf("enter array elements\n");
        for(i = 0 ; i < n ; i++)
        {
                scanf("%d",p+i);
        }

        printf("\nseperated even and odd elements of array : \n");
        for(i = 0 ; i < n ; i++)
        {
                for(j = i + 1 ; j < n ; j++)
                {
                        // here we check elements are even then seperate from odd numbers
                        if(*(p+i) % 2 == 0 && *(p+j) % 2 != 0)
                        {
                                temp = *(p+i);
                            *(p+i) = *(p+j);
                            *(p+j) = temp;
                        }
                }

        }

        for(i = 0 ; i < n ; i++)
        {
                printf("%d ",*(p+i));
        }

        getch();
}
```

```c
/*
* Que.14: Accept string from user and print the number of words using dynamic memory
allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       char *str = NULL;
       char ch;
       int no_of_char = 1;
       int i = 0;
       int cnt = 0;

       str = (char*)malloc(sizeof(char));
       *(str+0) = '\0';

       printf("please enter a string : ");
       do{
              scanf("%c",&ch);
              if(ch != '\n')
              {
                     no_of_char++;
                     str = (char*)realloc(str,no_of_char*sizeof(char));
                     *(str+i) = ch;
                     *(str+i+1) = '\0';
                     i++;
              }
       }while(ch != '\n');


       printf("your string is : \n");
       for(i = 0 ; *(str+i) != '\0' ; i++)
       {
              printf("%c",*(str+i));
       }

       i = 0;
       while(*(str+i) != '\0')
       {
              while(*(str+i) == ' ')
              {
                     i++;
              }

              if(*(str+i) != ' ' && *(str+i) != '\0')
              {
                     cnt++;
              }
```

```
        while(*(str+i) != ' ' && *(str+i) != '\0')
        {
            i++;
        }
    }

    printf("\nnumber of words in given string are : %d",cnt);

    getch();
}
```

```c
/*
* Que.15: Print unique elements in given array using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
	int *p;
	int n , i , j ;
	int cnt;

	printf("enter how many elements you want to insert : ");
	scanf("%d",&n);

	p = (int*)malloc(n*sizeof(int));

	printf("enter array elements\n");
	for(i = 0 ; i < n ; i++)
	{
		scanf("%d",p+i);
	}

	// here we check elements are unique or not
	printf("\nunique elements in array : \n");
	for(i = 0 ; i < n ; i++)
	{
		cnt = 0;
		for(j = 0 ; j < n ; j++)
		{
			if( *(p+i) == *(p+j) )
			{
				cnt++;
			}
		}
		if(cnt < 2)
		{
			printf("%d ",*(p+i));
		}

	}

	getch();
}
```

```c
/*
* Que.16: Accept string from user and replace goodname in mail using dynamic memory
allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char *gn = NULL;
        char ch;
        int no_of_char = 1;
        int i = 0,j,k = 0;
        int cnt = 0;

        gn = (char*)malloc(sizeof(char));
        *(gn+0) = '\0';

        printf("please enter a string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        gn = (char*)realloc(gn,no_of_char*sizeof(char));
                        *(gn+i) = ch;
                        *(gn+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');

        no_of_char = 1;
        i = 0;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter a string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');
        j = i;
        printf("\nraw string is :\n");
        for(i = 0 ; *(gn+i) != '\0' ; i++)
        {
                printf("%c",*(gn+i));
        }

        printf("\nexpected string is : \n");

        j = j + 6;
        for(i = 0 ; i < j , *(str+k) != '\0'; i++)
        {
                if(i >= 6)
                {
                        *(gn+i) = *(str+k);
                        k++;
                }
        }
        *(gn+i) = '\0';

        for(i = 0 ; *(gn+i) != '\0' ; i++)
        {
                printf("%c",*(gn+i));
        }

        getch();
}
```

```
/*
* Que.17: Insert new element in sorted array using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        int *p;
        int n , i , j ;
        int num; //new element
        int temp;

        printf("enter how many elements you want to insert : ");
        scanf("%d",&n);

        p = (int*)malloc(n*sizeof(int));

        printf("enter array elements\n");
        for(i = 0 ; i < n ; i++)
        {
                scanf("%d",p+i);
        }

        printf("\nenter new element you want to insert : ");
        scanf("%d",&num);

        // here we sort array
        for(i = 0 ; i < n ; i++)
        {
                for(j = i + 1 ; j < n ; j++)
                {
                        if(*(p+i) > *(p+j))
                        {
                            temp = *(p+i);
                            *(p+i) = *(p+j);
                            *(p+j) = temp;
                    }
                }
        }

        // here we find position to insert new element
        for(i = 0 ; i < n ; i++)
        {
                if(num < *(p+i))
                {
                        temp = i;
                        break;
                }
        }
```

```c
        // here we insert element logic
        for(i = n ; i >= temp ; i--)
        {
                *(p+i) = *(p+i-1);
                if(i == temp)
                {
                        *(p+i) = num;
                }
        }

        // new ele inserted array
        printf("\nnew element inserted array is : ");
        for(i = 0 ; i <= n ; i++)
        {
                printf("%d ",*(p+i));

        }

        getch();
}
```

```
/*
* Que.19: Delete element at desired position in array using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       int *p;
       int n , i , j ;
       int num; //delete position

       printf("enter how many elements you want to insert : ");
       scanf("%d",&n);

       p = (int*)malloc(n*sizeof(int));

       printf("enter array elements\n");
       for(i = 0 ; i < n ; i++)
       {
              scanf("%d",p+i);
       }

       printf("\nenter position to delete the element in array : ");
       scanf("%d",&num);

       //delete element at desired position
       for(i = num-1 ; i < n-1 ; i++)
       {
              *(p+i) = *(p+i+1);
       }

       printf("\ndeleted element array : ");
       for(i = 0 ; i < n-1 ; i++)
       {
              printf("%d ",*(p+i));
       }

       getch();
}
```

```c
/*
* Que.20: Accept string from user which contains char y to b using dynamic memory
allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char ch;
        int no_of_char = 1;
        int i = 0;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter a string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');

        printf("your string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }

        printf("\nexpected string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                if(*(str+i) == 'a' || *(str+i) == 'z')
                {
                        break;
                }
                printf("%c",*(str+i));
        }
        getch();
}
```

```c
/*
* Que.21: Accept string from user and count small capital n digits n spaces using dynamic
memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char ch;
        int no_of_char = 1;
        int i = 0;
        int cntsc = 0;
        int cntcc = 0;
        int cntd = 0;
        int cnts = 0;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter a string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');

        printf("your string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }

        i = 0;
        while(*(str+i) != '\0')
        {
                if(*(str+i) == ' ')
                {
                        cnts++;
                }

                else if(*(str+i) >= 'a' && *(str+i) <= 'z')
                {
                        cntsc++;
                }

                else if(*(str+i) >= 'A' && *(str+i) <= 'Z')
                {
                        cntcc++;
                }

                else if(*(str+i) >= '1' && *(str+i) <= '9')
                {
                        cntd++;
                }
                i++;
        }

        printf("\nnumbers of small alphabets are : %d\n",cntsc);
        printf("numbers of capital alphabets are : %d\n",cntcc);
        printf("numbers of digits are : %d\n",cntd);
        printf("numbers of spaces are : %d\n",cnts);

        getch();
}
```

```c
/*
* Que.22: Print minimum and maximum number in array using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        int *p;
        int n , i;
        int min , max;

        printf("enter how many elements you want to insert : ");
        scanf("%d",&n);

        p = (int*)malloc(n*sizeof(int));

        printf("enter array elements\n");
        for(i = 0 ; i < n ; i++)
        {
                scanf("%d",p+i);
        }

        //minimum number
        min = *(p+0);
        for(i=0 ; i < n ; i++)
        {
                if(*(p+i) < min)
                {
                        min = *(p+i);
                }
        }
        printf("\nminimum number is %d",min);

    // maximum number
        max = *(p+0);
        for(i=0 ; i < n ; i++)
        {
                if(*(p+i) > max)
                {
                        max = *(p+i);
                }
        }
        printf("\nmaximum number is %d",max);

        getch();
}
```

```c
/*
* Que.23: Accept string from user and count white spaces using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char ch;
        int no_of_char = 1;
        int i = 0;
        int cnt = 0;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter a string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');

        printf("your string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }

        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                if(*(str+i) == ' ')
                {
                        cnt++;
                }
        }
        printf("\nnumbers of spaces are : %d\n",cnt);

        getch();
}
```

```c
/*
* Que.24: Find second highest element in array using dynamically
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        int *p;
        int n , i;
        int max;
        int max2;

        printf("enter how many elements you want to insert : ");
        scanf("%d",&n);

        p = (int*)malloc(n*sizeof(int));

        printf("enter array elements\n");
        for(i = 0 ; i < n ; i++)
        {
                scanf("%d",p+i);
        }

        printf("array elements are\n");
        for(i = 0 ; i < n ; i++)
        {
                printf("%d ",*(p+i));
        }

        //find second highest element
        max = *(p+0);
        for(i = 0 ; i < n ; i++)
        {
                if(*(p+i) > max)
                {
                        max2 = max;
                        max = *(p+i);
                }
                else if(*(p+i) < max && *(p+i) > max2)
                {
                        max2 = *(p+i);
                }
        }

        printf("\nsecond highest element is : %d",max2);
        getch();
}
```

```c
/*
* Que.25: Accept string from user and print number of words of even and odd length using
dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       char *str = NULL;
       char ch;
       int no_of_char = 1;
       int i = 0;
       int cnt;
       int ecnt = 0;
       int ocnt = 0;

       str = (char*)malloc(sizeof(char));
       *(str+0) = '\0';

       printf("please enter a string : ");
       do{
              scanf("%c",&ch);
              if(ch != '\n')
              {
                     no_of_char++;
                     str = (char*)realloc(str,no_of_char*sizeof(char));
                     *(str+i) = ch;
                     *(str+i+1) = '\0';
                     i++;
              }
       }while(ch != '\n');

       printf("your string is : \n");
       for(i = 0 ; *(str+i) != '\0' ; i++)
       {
              printf("%c",*(str+i));
       }

       i = 0;
       while(*(str+i) != '\0')
       {
              while(*(str+i) == ' ')
              {
                     i++;
              }

              cnt = 0;
              while(*(str+i) != ' ' && *(str+i) != '\0')
              {
                     cnt++;
                     i++;
              }

        if(cnt != 0)
              {
              if(cnt % 2 == 0)
              {
                     ecnt++;
              }
              else if(cnt % 2 != 0)
              {
                     ocnt++;
              }
              }

       }
       printf("\neven lenght words are : %d\n",ecnt);
       printf("odd lenght words are : %d",ocnt);
       getch();
}
```

```c
/*
* Que.27: Accept string from user and print last words of given sentence using dynamic
memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
	char *str = NULL;
	char *s[20];
	char ch;
	int no_of_char = 1;
	int i = 0,j = 0;
	int cnt;

	str = (char*)malloc(sizeof(char));
	*(str+0) = '\0';

	printf("please enter a string : ");
	do{
		scanf("%c",&ch);
		if(ch != '\n')
		{
			no_of_char++;
			str = (char*)realloc(str,no_of_char*sizeof(char));
			*(str+i) = ch;
			*(str+i+1) = '\0';
			i++;
		}
	}while(ch != '\n');

	printf("your string is : \n");
	for(i = 0 ; *(str+i) != '\0' ; i++)
	{
		printf("%c",*(str+i));
	}
	cnt = i;

	//here we stores last word reversly in other string
	for(i = cnt ; *(str+i) != ' ' ; i--)
	{
		ch = *(str+i);
		s[j] = ch;
		j++;
	}

	//here we print last reverse word reversly to get expected word
	printf("\nlast word is : ");
	for(i = j-1 ; i >= 0 ; i--)
	{
```

```
                printf("%c",s[i]);
        }

        getch();
}
```

```
/*
* Que.29: Accept string and position from user and print words of given sentence using
dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char *s[20];
        char ch;
        int no_of_char = 1;
        int i = 0;
        int pw;
        int cnt = 0;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter a string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');

        printf("your string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }

        printf("\nenter position of word which you want : ");
        scanf("%d",&pw);

        i = 0;
        while(*(str+i) != '\0')
        {
                while(*(str+i) == ' ')
                {
                        i++;
                }

                if(*(str+i) != ' ')
```

```
              {
                     cnt++;
              }

              while(*(str+i) != ' ' && *(str+i) != '\0')
              {
                     if(pw == cnt)
                     {
                            printf("%c",*(str+i));
                     }
                     i++;
              }
       }

       if(pw > cnt)
              printf("given string contains only %d words",cnt);

       getch();
}


/*
* Que.30: Print alternate elements in array dynamically
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       int *p;
       int n , i;

       printf("enter how many elements you want to insert : ");
       scanf("%d",&n);

       p = (int*)malloc(n*sizeof(int));

       printf("enter array elements\n");
       for(i = 0 ; i < n ; i++)
       {
              scanf("%d",p+i);
       }

       printf("array elements are \n");
       for(i = 0 ; i < n ; i = i + 2)
       {
              printf("%d ",*(p+i));
       }

       getch();
}
```

```c
/*
* Que.31: Accept string from user and convert upper case to lower case using dynamic
memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       char *str = NULL;
       char *s[20];
       char ch;
       int no_of_char = 1;
       int i = 0;

       str = (char*)malloc(sizeof(char));
       *(str+0) = '\0';

       printf("please enter a string : ");
       do{
              scanf("%c",&ch);
              if(ch != '\n')
              {
                     no_of_char++;
                     str = (char*)realloc(str,no_of_char*sizeof(char));
                     *(str+i) = ch;
                     *(str+i+1) = '\0';
                     i++;
              }
       }while(ch != '\n');

       printf("your string is : \n");
       for(i = 0 ; *(str+i) != '\0' ; i++)
       {
              printf("%c",*(str+i));
       }

       printf("\nconverted string is :\n");
       i = 0;
       while(*(str+i) != '\0')
       {
              if(*(str+i) >= 'A' && *(str+i) <= 'Z')
              {
                     *(str+i) = *(str+i) + 32;
              }
              printf("%c",*(str+i));
              i++;
       }

       getch();
}
```

```c
/*
* Que.32: Find 2 elements that difference bet them largest in array using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       int *p;
       int n , i;
       int min , max;

       printf("enter how many elements you want to insert : ");
       scanf("%d",&n);

       p = (int*)malloc(n*sizeof(int));

       printf("enter array elements\n");
       for(i = 0 ; i < n ; i++)
       {
              scanf("%d",p+i);
       }

       //minimum number
       min = *(p+0);
       for(i=0 ; i < n ; i++)
       {
              if(*(p+i) < min)
              {
                     min = *(p+i);
              }
       }

       // maximum number
       max = *(p+0);
       for(i=0 ; i < n ; i++)
       {
              if(*(p+i) > max)
              {
                     max = *(p+i);
              }
       }
       printf("\n%d and %d are two number differnce %d is largest",max,min,max-min);

       getch();
}
```

```c
/*
* Que.33: Accept string from user and toggle the cases using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       char *str = NULL;
       char ch;
       int no_of_char = 1;
       int i = 0;

       str = (char*)malloc(sizeof(char));
       *(str+0) = '\0';

       printf("please enter a string : ");
       do{
              scanf("%c",&ch);
              if(ch != '\n')
              {
                     no_of_char++;
                     str = (char*)realloc(str,no_of_char*sizeof(char));
                     *(str+i) = ch;
                     *(str+i+1) = '\0';
                     i++;
              }
       }while(ch != '\n');

       printf("your string is : \n");
       for(i = 0 ; *(str+i) != '\0' ; i++)
       {
              printf("%c",*(str+i));
       }

       printf("\ntoggle string is :\n");
       i = 0;
       while(*(str+i) != '\0')
       {
              if(*(str+i) >= 'A' && *(str+i) <= 'Z')
              {
                     *(str+i) = *(str+i) + 32;
              }

              else if(*(str+i) >= 'a' && *(str+i) <= 'z')
              {
                     *(str+i) = *(str+i) - 32;
              }

              i++;
       }
```

```c
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }

        getch();
}




/*
* Que.34: Accept string from user and store their ascii values in int array using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        int *str2;
        char ch;
        int no_of_char = 1;
        int i = 0;
        int i2;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter a string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');
        i2 = i;

        str2 = (int*)malloc(i2*sizeof(int));

        printf("your string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
```

```
       }

       for(i = 0 ; *(str+i) != '\0' ; i++)
    {
              *(str2+i) = *(str+i);
       }
       *(str2+i) = '\0';

       printf("\nascii values of string are : \n");

       for(i = 0 ; *(str2+i) != '\0' ; i++)
       {
              printf("%c = %d ",*(str2+i),*(str2+i));
       }

       getch();
}
```

```
/*
* Que.35: Accept two strings from user and check anagram or not using dynamic memory
allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       char *str = NULL;
       char *str2 = NULL;
       char ch;
       int no_of_char = 1;
       int i = 0,j;
       char temp;
       int cnt1 = 0;
       int cnt2 = 0;
       int cnt = 0;

       str = (char*)malloc(sizeof(char));
       *(str+0) = '\0';

       printf("please enter 1st string : ");
       do{
              scanf("%c",&ch);
              if(ch != '\n')
              {
                     no_of_char++;
                     str = (char*)realloc(str,no_of_char*sizeof(char));
                     *(str+i) = ch;
                     *(str+i+1) = '\0';
                     i++;
              }
       }while(ch != '\n');

       str2 = (char*)malloc(sizeof(char));
       *(str2+0) = '\0';

       no_of_char = 1;
       i = 0;
       printf("please enter 2nd string : ");
       do{
              scanf("%c",&ch);
              if(ch != '\n')
              {
                     no_of_char++;
                     str2 = (char*)realloc(str2,no_of_char*sizeof(char));
                     *(str2+i) = ch;
                     *(str2+i+1) = '\0';
                     i++;
              }
```

```
}while(ch != '\n');

printf("\nyour 1st string is : \n");
for(i = 0 ; *(str+i) != '\0' ; i++)
{
       printf("%c",*(str+i));
}
cnt1 = i;

printf("\nyour 2nd string is : \n");
for(i = 0 ; *(str2+i) != '\0' ; i++)
{
       printf("%c",*(str2+i));
}
cnt2 = i;

if(cnt1 != cnt2)
       printf("\nboth strings are not equal in length");

if(cnt1 == cnt2)
{
//logic for str1 sorting
for(i = 0 ; i <= cnt1 ; i++)
{
       for(j = i + 1 ; j <= cnt1 ; j++)
       {
              if(*(str+i) > *(str+j))
              {
                     temp = *(str+i);
                  *(str+i) = *(str+j);
                  *(str+j) = temp;
              }
       }
}

//logic for str2 sorting
for(i = 0 ; i <= cnt2 ; i++)
{
       for(j = i + 1 ; j <= cnt2 ; j++)
       {
              if(*(str2+i) > *(str2+j))
              {
                     temp = *(str2+i);
                  *(str2+i) = *(str2+j);
                  *(str2+j) = temp;
              }
       }
}

i = 0;
while(i < cnt1)
{
       if(*(str+i) == *(str2+i))
       {
              cnt++;
       }
       i++;
}
```

```
        //printf("%d\n",cnt);

        if(cnt == cnt1)
                printf("\nstrings are anagram");
        else
                printf("\nstrings are not anagram\n");
        }

        getch();
}
```

```
/*
* Que.36: Stores squares of numbers in same array using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        int *p;
        int n , i ;

        printf("enter how many elements you want to insert : ");
        scanf("%d",&n);

        p = (int*)malloc(n*sizeof(int));

        printf("enter array elements\n");
        for(i = 0 ; i < n ; i++)
        {
                scanf("%d",p+i);
        }

        printf("squred number array elements are : \n");
        for(i = 0 ; i < n ; i++)
        {
                *(p+i) = *(p+i) * *(p+i);
                printf("%d ",*(p+i));
        }

        getch();
}
```

```
/*
* Que.37: Accept string from user and copy into another string using dynamic memory
allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       char *str = NULL;
       char *str2 = NULL;
       char ch;
       int no_of_char = 1;
       int i = 0;
       char temp;

       str = (char*)malloc(sizeof(char));
       *(str+0) = '\0';

       printf("please enter a string : ");
       do{
              scanf("%c",&ch);
              if(ch != '\n')
              {
                     no_of_char++;
                     str = (char*)realloc(str,no_of_char*sizeof(char));
                     *(str+i) = ch;
                     *(str+i+1) = '\0';
                     i++;
              }
       }while(ch != '\n');


       printf("your string is : \n");
       for(i = 0 ; *(str+i) != '\0' ; i++)
       {
              printf("%c",*(str+i));
       }
       temp = i;

       str2 = (char*)malloc(temp*sizeof(char));

       printf("\ncopied string is : \n");
       for(i = 0 ; *(str+i) != '\0' ; i++)
       {
              *(str2+i) = *(str+i);
       }
       *(str2+i) = '\0';

       for(i = 0 ; *(str2+i) != '\0' ; i++)
       {
```

```
                printf("%c",*(str2+i));
        }
        getch();
}
```

```c
/*
* Que.38: Accept string from user and copy first n elements into another string using
dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char *str2 = NULL;
        char ch;
        int no_of_char = 1;
        int i = 0;
        char temp;
        int s;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter a string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');


        printf("your string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }
        temp = i;

        str2 = (char*)malloc(temp*sizeof(char));

        printf("\nenter position of word which you want : ");
        scanf("%d",&s);

        if(s > temp)
                printf("given string have only %d positions !!",temp);

        for(i = 0 ; i <= s ; i++)
        {
                *(str2+i) = *(str+i);
        }
        *(str2+i) = '\0';

        printf("your copied string upto first n char is : \n");
        for(i = 0 ; *(str2+i) != '\0' ; i++)
        {
                printf("%c",*(str2+i));
        }

        getch();
}
```

```
/*
* Que.39: Find 2 elements such their sum is closest to given number in array using
dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       int *p;
       int n , i , j;
       int temp;
       int min;

       printf("enter how many elements you want to insert : ");
       scanf("%d",&n);

       p = (int*)malloc(n*sizeof(int));

       printf("enter array elements\n");
       for(i = 0 ; i < n ; i++)
       {
              scanf("%d",p+i);
       }

       // here we sort array
       for(i = 0 ; i < n ; i++)
       {
              for(j = i + 1 ; j < n ; j++)
              {
                     if(*(p+i) > *(p+j))
                     {
                         temp = *(p+i);
                         *(p+i) = *(p+j);
                         *(p+j) = temp;
                  }
              }
       }

       min = *(p+0) + *(p+1);

       printf("\n%d and %d elements such their sum %d is closest to given
number",*(p+0),*(p+1),min);

       getch();
}
```

```
/*
* Que.40: Accept string from user and copy last n elements into another string using
dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char *str2 = NULL;
        char ch;
        int no_of_char = 1;
        int i = 0,j;
        char temp;
        int s;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter a string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');


        printf("your string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }
        temp = i;

        str2 = (char*)malloc(temp*sizeof(char));

        printf("\nenter position of word which you want : ");
        scanf("%d",&s);

        if(s > temp)
                printf("given string have only %d positions !!",temp);
        j = 0;
        for(i = s ; i <= temp ; i++)
        {
                *(str2+j) = *(str+i);
                j++;
        }
        *(str2+j) = '\0';

        printf("your copied string upto last n char is : \n");
        for(i = 0 ; *(str2+i) != '\0' ; i++)
        {
                printf("%c",*(str2+i));
        }

        getch();
}
```

```
/*
* Que.41: Find given int X appears more than N/2 times in sorted array of N int using
dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        int *p;
        int n , i , j ;
        int *x;
        int temp;
        int cnt = 0;

        printf("enter how many elements you want to insert : ");
        scanf("%d",&n);

        p = (int*)malloc(n*sizeof(int));

        printf("enter array elements\n");
        for(i = 0 ; i < n ; i++)
        {
                scanf("%d",p+i);
        }

        printf("\nenter a number you want to find more than %d times\n",n/2);

        x = (int*)malloc(sizeof(int));

        scanf("%d",x);

        // here we sort array
        for(i = 0 ; i < n ; i++)
        {
                for(j = i + 1 ; j < n ; j++)
                {
                        if(*(p+i) > *(p+j))
                        {
                            temp = *(p+i);
                            *(p+i) = *(p+j);
                            *(p+j) = temp;
                    }
                }
        }

        // here we count element is how many times insertd in array
        for(i = 0 ; i < n ; i++)
        {
                if(*(p+i) == *x)
                        cnt++;
```

```
    }

    if(cnt == 0)
        printf("%d is not a given array element",*x);
    else if(cnt > n/2)
        printf("%d element appears %d times i.e. more than %d times",*x,cnt,n/2);
    else
        printf("%d element appears less than %d times",*x,n/2);


    getch();
}
```

```c
/*
* Que.42: Accept two strings from user and append 2nd after 1st using dynamic memory
allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char *str2 = NULL;
        char ch;
        int no_of_char = 1;
        int i = 0,j;
        int cnt1 = 0;
        int cnt2 = 0;
        int cnt = 0;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter 1st string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');

        str2 = (char*)malloc(sizeof(char));
        *(str2+0) = '\0';

        no_of_char = 1;
        i = 0;
        printf("please enter 2nd string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str2 = (char*)realloc(str2,no_of_char*sizeof(char));
                        *(str2+i) = ch;
                        *(str2+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');
        printf("\nyour 1st string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }
        cnt1 = i;

        printf("\nyour 2nd string is : \n");
        for(i = 0 ; *(str2+i) != '\0' ; i++)
        {
                printf("%c",*(str2+i));
        }
        cnt2 = i;
        cnt = cnt1+cnt2;

        str = (char*)realloc(str,cnt*sizeof(char));

        j = 0;
        for(i = cnt1 ; i <= cnt ; i++)
        {
                *(str+i) = *(str2+j);
                j++;
        }
        *(str+i) = '\0';

        printf("\nappend string is\n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }
        getch();
}
```

```
/*
* Que.44: Append 2nd string after 1st upto n char using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char *str2 = NULL;
        char ch;
        int no_of_char = 1;
        int i = 0,j;
        int cnt1 = 0;
        int cnt2 = 0;
        int cnt;
        int s;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter 1st string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');

        str2 = (char*)malloc(sizeof(char));
        *(str2+0) = '\0';

        no_of_char = 1;
        i = 0;
        printf("please enter 2nd string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str2 = (char*)realloc(str2,no_of_char*sizeof(char));
                        *(str2+i) = ch;
                        *(str2+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');
        printf("\nyour 1st string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }
        cnt1 = i;

        printf("\nyour 2nd string is : \n");
        for(i = 0 ; *(str2+i) != '\0' ; i++)
        {
                printf("%c",*(str2+i));
        }
        cnt2 = i;

        printf("\nenter a position for 2nd string you want to append : ");
        scanf("%d",&s);

        cnt = cnt1 + s;

        str = (char*)realloc(str,cnt*sizeof(char));

        j = 0;
        for(i = cnt1 ; i < cnt ; i++)
        {
                *(str+i) = *(str2+j);
                j++;
        }
        *(str+i) = '\0';

        printf("\nappend string is\n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }

        getch();
}
```

```c
/*
* Que.45: Check both strings are equal if not then print diff bet 1st mismatch char using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char *str2 = NULL;
        char ch;
        int no_of_char = 1;
        int i = 0;
        int cnt1 = 0;
        int cnt2 = 0;
        int cnt = 0;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter 1st string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');

        str2 = (char*)malloc(sizeof(char));
        *(str2+0) = '\0';

        no_of_char = 1;
        i = 0;
        printf("please enter 2nd string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str2 = (char*)realloc(str2,no_of_char*sizeof(char));
                        *(str2+i) = ch;
                        *(str2+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');
        printf("\nyour 1st string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }
        cnt1 = i;

        printf("\nyour 2nd string is : \n");
        for(i = 0 ; *(str2+i) != '\0' ; i++)
        {
                printf("%c",*(str2+i));
        }
        cnt2 = i;

        for(i = 0 ; i != cnt1 ; i++)
        {
                if(*(str+i) == *(str2+i))
                        cnt++;
        }
        if(i == cnt && cnt1 == cnt2)
                printf("\nboth strings are equal");
        else
        {
        i = 0;
        cnt2 = 0;
        while(*(str+i) == *(str2+i))
        {
                cnt2++;
                i++;
        }
        i = *(str+cnt2) - *(str2+cnt2);
        if(-i == -i)
                i = -i;
        printf("\ndifference between first mismatch char %c and %c is 
%d",*(str+cnt2),*(str2+cnt2),i);
        }

        getch();
}
```

```c
/*
* Que.46: Check both strings are equal upto 2nd string n char if not then print diff bet
1st mismatch char using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char *str2 = NULL;
        char ch;
        int no_of_char = 1;
        int i = 0;
        int cnt1 = 0;
        int cnt2 = 0;
        int s,min;
        int cnt = 0;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter 1st string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');

        str2 = (char*)malloc(sizeof(char));
        *(str2+0) = '\0';

        no_of_char = 1;
        i = 0;
        printf("please enter 2nd string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str2 = (char*)realloc(str2,no_of_char*sizeof(char));
                        *(str2+i) = ch;
                        *(str2+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');

        printf("\nyour 1st string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }
        cnt1 = i;

        printf("\nyour 2nd string is : \n");
        for(i = 0 ; *(str2+i) != '\0' ; i++)
        {
                printf("%c",*(str2+i));
        }
        cnt2 = i;

        printf("enter n th position : ");
        scanf("%d",&s);

        if(cnt1 > cnt2)
                min = cnt2;
        else
                min = cnt1;

        if(s > min)
                printf("you can check char only %d positions",min);
        else
        {
        for(i = 0 ; i < s ; i++)
        {
                if(*(str+i) == *(str2+i))
                        cnt++;
        }
        if(i == cnt)
                printf("both strings are equal");
        else
        {
        i = 0;
        cnt2 = 0;
        while(*(str+i) == *(str2+i))
        {
                cnt2++;
                i++;
        }
        i = *(str+cnt2) - *(str2+cnt2);
        printf("difference between first mismatch char %c and %c is 
%d",*(str+cnt2),*(str2+cnt2),i);
        }

        }

        getch();
}
```

```c
/*
* Que.48: Check both strings are equal or not without case sensitive using dynamic memory
allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       char *str = NULL;
       char *str2 = NULL;
       char ch;
       int no_of_char = 1;
       int i = 0;
       int cnt1 = 0;
       int cnt2 = 0;
       int cnt = 0;

       str = (char*)malloc(sizeof(char));
       *(str+0) = '\0';

       printf("please enter 1st string : ");
       do{
              scanf("%c",&ch);
              if(ch != '\n')
              {
                     no_of_char++;
                     str = (char*)realloc(str,no_of_char*sizeof(char));
                     *(str+i) = ch;
                     *(str+i+1) = '\0';
                     i++;
              }
       }while(ch != '\n');

       str2 = (char*)malloc(sizeof(char));
       *(str2+0) = '\0';

       no_of_char = 1;
       i = 0;
       printf("please enter 2nd string : ");
       do{
              scanf("%c",&ch);
              if(ch != '\n')
              {
                     no_of_char++;
                     str2 = (char*)realloc(str2,no_of_char*sizeof(char));
                     *(str2+i) = ch;
                     *(str2+i+1) = '\0';
                     i++;
              }
       }while(ch != '\n');
```

```
    printf("\nyour 1st string is : \n");
    for(i = 0 ; *(str+i) != '\0' ; i++)
    {
        printf("%c",*(str+i));
    }
    cnt1 = i;

    printf("\nyour 2nd string is : \n");
    for(i = 0 ; *(str2+i) != '\0' ; i++)
    {
        printf("%c",*(str2+i));
    }
    cnt2 = i;

    if(cnt1 != cnt2)
        printf("both strings are unequal due to lenght");
    else
    {
        for(i = 0 ; i < cnt1 ; i++)
        {
        if(*(str+i) >= 'A' && *(str+i) <= 'Z')
        {
            *(str+i) = *(str+i)+32;
        }
        }

        for(i = 0 ; i < cnt2 ; i++)
        {
        if(*(str2+i) >= 'A' && *(str2+i) <= 'Z')
        {
            *(str2+i) = *(str2+i)+32;
        }
        }

    for(i = 0 ; i < cnt1 ; i++)
    {
        if(*(str+i) == *(str2+i))
            cnt++;
    }
    if(i == cnt)
        printf("\nboth strings are equal");
    else
    {
    i = 0;
    cnt2 = 0;
    while(*(str+i) == *(str2+i))
    {
        cnt2++;
        i++;
    }
    i = *(str+cnt2) - *(str2+cnt2);
    printf("\ndifference between first mismatch char is %d",i);
    }

    }
    getch();
}
```

```
/*
* Que.49: Reverse the string upto n char using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       char *str = NULL;
       char ch;
       int no_of_char = 1;
       int i = 0;
       int cnt;
       int s;

       str = (char*)malloc(sizeof(char));
       *(str+0) = '\0';

       printf("please enter a string : ");
       do{
              scanf("%c",&ch);
              if(ch != '\n')
              {
                     no_of_char++;
                     str = (char*)realloc(str,no_of_char*sizeof(char));
                     *(str+i) = ch;
                     *(str+i+1) = '\0';
                     i++;
              }
       }while(ch != '\n');

       printf("\nyour string is : \n");
       for(i = 0 ; *(str+i) != '\0' ; i++)
       {
              printf("%c",*(str+i));
       }
       cnt = i;

       printf("\nenter a position you want reverse the given string : ");
       scanf("%d",&s);

       if(s > cnt)
              printf("lenght of string is only %d remaing string prints may be garbage 
value",cnt);

       for(i = s-1 ; i >= 0 ; i--)
       {
              printf("%c",*(str+i));
       }
       for(i = s ; *(str+i) != '\0' ; i++)
       {
        printf("%c",*(str+i));
    }

    getch();
}
```

```
/*
* Que.50: Reverse the string within range using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char ch;
        int no_of_char = 1;
        int i = 0;
        int cnt;
        int s1,s2;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter a string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');

        printf("\nyour string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }
        cnt = i;

        printf("\nenter range 1st : ");
        scanf("%d",&s1);
        printf("enter range 2nd : ");
        scanf("%d",&s2);

        if(s2 > cnt)
                printf("lenght of string is only %d remaing string prints may be garbage
value",cnt);

        for(i = 0 ; i <= s1-1 ; i++)
        {
                printf("%c",*(str+i));
        }
```

```c
        for(i = s2 ; i >= s1 ; i--)
        {
                printf("%c",*(str+i));
        }
        for(i = s2+1 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }

        getch();
}
```

```c
/*
* Que.51: Find max sum of subsequent numbers in array using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        int *p;
        int n , i;
        int max2 , max ;

        printf("enter how many elements you want to insert : ");
        scanf("%d",&n);

        p = (int*)malloc(n*sizeof(int));

        printf("enter array elements\n");
        for(i = 0 ; i < n ; i++)
        {
                scanf("%d",p+i);
        }

        max = *(p+0) + *(p+1);
        for(i = 0 ; i < n ; i++)
        {
                max2 = *(p+i) + *(p+i+1);
                if(max < max2)
                {
                        max = max2;
                }
        }

        printf("\nmaximum sum of subsequent numbers is %d",max);

        getch();
}
```

```
/*
* Que.52: Reverse the even lenght words in string using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       char *str = NULL;
       char ch;
       int no_of_char = 1;
       int i = 0,j = 0 , k = 0;
       int cnt = 0;

       str = (char*)malloc(sizeof(char));
       *(str+0) = '\0';

       printf("please enter a string : ");
       do{
              scanf("%c",&ch);
              if(ch != '\n')
              {
                     no_of_char++;
                     str = (char*)realloc(str,no_of_char*sizeof(char));
                     *(str+i) = ch;
                     *(str+i+1) = '\0';
                     i++;
              }
       }while(ch != '\n');

       printf("\nyour string is : \n");
       for(i = 0 ; *(str+i) != '\0' ; i++)
       {
              printf("%c",*(str+i));
       }
       printf("\n");

       i = 0;
       while(*(str+i) != '\0')
       {
              //logic for skip spaces
              while(*(str+i) == ' ')
              {
                     i++;
              }
              j = i;
              cnt = 0;
              //if string is char then count how many char in one word
              while(*(str+i) != ' ' && *(str+i) != '\0')
              {
                     cnt++;
```

```
                        i++;
                }
                k = i-1;
                //if count of word is even print reverse the word
         if(cnt != 0)
                {
                if(cnt % 2 == 0)
                {
                        while(k >= j && *(str+k) != '\0')
                {
                        printf("%c",*(str+k));
                        k--;
                }
                        printf(" ");
                }
                //if count of word is odd print the word as it is
                else if(cnt % 2 != 0)
                {
                        while(j <= k && *(str+j) != '\0')
                {
                        printf("%c",*(str+j));
                        j++;
                }
                        printf(" ");
                }
                }

        }

        getch();
}
```

```
/*
* Que.53: Check string pallindrome or not using dynamic memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char *str2 = NULL;
        char ch;
        int no_of_char = 1;
        int i = 0,j = 0 , s = 0;
        int cnt = 0;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter a string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');

        printf("\nyour string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }
        cnt = i;

        str2 = (char*)malloc(cnt*sizeof(char));

        //here we copy given string into another string
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                *(str2+i) = *(str+i);
        }
        *(str2+i) = '\0';

        //here logic of pallindrome
        j = 0;
        for(i = cnt-1 ; i >= 0 ; i--)
        {
```

```
        if(*(str2+j) == *(str+i))
        {
            s++;
        }
        j++;
    }

    //here we check string is pallindrome or not
    if(s == j)
        printf("\nstring is pallindrome");
    else
        printf("\nstring is not pallindrome");

    getch();
}
```

```
/*
* Que.54: Count number of elements greather ,less or equal to zero of array using dynamic
memory allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
       int *p;
       int n , i;
       int g = 0;
       int l = 0;
       int z = 0;

       printf("enter how many elements you want to insert : ");
       scanf("%d",&n);
       p = (int*)malloc(n*sizeof(int));

       printf("enter array elements\n");
       for(i = 0 ; i < n ; i++)
       {
              scanf("%d",p+i);
       }
       printf("array elements are\n");
       for(i = 0 ; i < n ; i++)
       {
              printf("%d ",*(p+i));
       }
       for(i = 0 ; i < n ; i++)
       {
              // greater than zero
              if(*(p+i) > 0)
              {
                     g++;
              }
              // less than zero
              else if(*(p+i) < 0)
              {
                     l++;
              }
              // count if number is zero
              else
                     z++;

       }
       printf("\ngreater than zero elements are %d",g);
       printf("\nless than zero elements are %d",l);
       printf("\nequal to zero elements are %d",z);
       getch();
}
```

```
/*
* Que.55: Count words, alphabets and spaces in given string using dynamic memory
allocation
* owner : Shreya Kailas Saskar
* batch : PPA9
*/


// solution :

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

void main()
{
        char *str = NULL;
        char ch;
        int no_of_char = 1;
        int i = 0;
        int cnt = 0;

        str = (char*)malloc(sizeof(char));
        *(str+0) = '\0';

        printf("please enter a string : ");
        do{
                scanf("%c",&ch);
                if(ch != '\n')
                {
                        no_of_char++;
                        str = (char*)realloc(str,no_of_char*sizeof(char));
                        *(str+i) = ch;
                        *(str+i+1) = '\0';
                        i++;
                }
        }while(ch != '\n');

        printf("\nyour string is : \n");
        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                printf("%c",*(str+i));
        }

        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
                if(*(str+i) == ' ')
                {
                        cnt++;
                }
        }
        printf("\nnumbers of spaces are : %d\n",cnt);

        cnt  = 0;

        for(i = 0 ; *(str+i) != '\0' ; i++)
        {
```

```c
                if(*(str+i) >= 'a' && *(str+i) <= 'z' || *(str+i) >= 'A' && *(str+i) <= 'Z')
                {
                        cnt++;
                }
        }
        printf("number of characters in given string are %d\n",cnt);

        cnt = 0;
        i = 0;

        while(*(str+i) != '\0')
        {
                while(*(str+i) == ' ')
                {
                        i++;
                }

                if(*(str+i) != ' ' && *(str+i) != '\0')
                {
                        cnt++;
                }

                while(*(str+i) != ' ' && *(str+i) != '\0')
                {
                        i++;
                }
        }
        printf("number of words in given string are : %d",cnt);
        getch();
}
```