https://goo.gl/5Vwyeo

1. Write a function *void calc*(*int *num1*, *int *num2*, int *res, *char *c*) that will take pointers to two integers, performs addition, subtraction, multiplication, increment or decrement according to the character pointed by the pointer *c* and store the result in the location pointed by *res*.

| Input | Output |
|---|---|
| 5 3 a<br>9 7 s<br>2 7 m | 8<br>2<br>14 |

2. Write a function that will exchange the contents of two variables.

| Input | Output |
|---|---|
| 3 5<br>6 8<br>0 4 | 5 3<br>8 6<br>4 0 |

3. Write a program to sort a list of integers. You have to perform pointer arithmetic. Use dynamic memory allocation and do not use array anywhere in your program. If you need to swap to element use the function you have implemented in problem 2.

| Input | Output |
|---|---|
| 5<br>8 6 -1 0 4 | -1 0 4 6 8 |

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

void swap(int *a, int *b){
   int t;
   t=*a;
   *a=*b;
   *b=t;
}

void sort(int *a, int n){

   int i,j;

   for(i=0;i<n-1;i++)
      for(j=0;j<n-i-1;j++)
         if(*(a+j)>*(a+j+1))swap((a+j),(a+j+1));
}

int main(){

   int *a,n,i;
   scanf("%d",&n);

   // dynamic memory allocation
   a= (int *)malloc(n*sizeof(int));

   for(i=0;i<n;i++){
      scanf("%d",a+i);
   }

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

   free(a);

   return 0;
}
```

4. Write a function *int strlen*(*char *str*) to compute length of a string. Do not declare any variable of primitive data type in your function, but you can use additional pointer variable(s).

| Input | Output |
|---|---|
| abcdef<br>xyz<br>aaaaaaa | 6<br>3<br>7 |

```
#include<stdio.h>
#include<stdlib.h>
int strlen(char *s){
   char *a=s;
   while(*a!='\0'){
      a++;
   }

   return (a-s);

}

int main(){
   char str[50];
   scanf("%s",str);

   printf("%d\n",strlen(str));
   return 0;
}
```

5. Write a function *char * strrev*(*char *str*) that will return the reverse of a string.

| Input | Output |
|---|---|
| abcdef<br>aaaa<br>xyz | fedcba<br>aaaa<br>zyx |

6. Write a program that will dynamically allocate memory for a 2D-array, take input into that array and print the array. You have to use pointer arithmetic for iterating through the array.

| Input | Output |
|---|---|
| 4 3<br>1 2 3<br>2 3 4<br>3 4 5<br>4 5 6<br><br>2 4<br>9 8 7 6<br>10 2 7 3 | 1 2 3<br>2 3 4<br>3 4 5<br>4 5 6<br><br>9 8 7 6<br>10 2 7 3 |

```
#include<stdio.h>
#include<stdlib.h>
int main(){
   int **a;
   int r,c,i,j;
   scanf("%d %d",&r,&c);

   a= (int **) malloc(r* sizeof(int *));

   for(i=0;i<r;i++){

      *(a+i)= (int *)malloc(c*sizeof(int));
   }

   for(i=0;i<r;i++){
      for(j=0;j<c;j++)
         scanf("%d",*(a+i)+j);
   }

   for(i=0;i<r;i++){
      for(j=0;j<c;j++)
         printf("%d ",*(*(a+i)+j));
      printf("\n");
```

```
    }
    for(i=0;i<r;i++) free(*(a+i));
    free(a);
    return 0;
}
```

7. struct point{
        char ch;
        double x;
        double y;
   };
   Dynamically allocate memory for an array of type struct.