

Practical exercises. Dynamic Memory Allocation in C Language

**Ex.1: Dynamic memory allocation for 1-D array.
Input and output of dynamic 1-D array**

```
#include <stdio.h>

#include<stdlib.h>

int main( )
{
    int *A, n, i;
    clrscr( );
    printf("\n Enter number of elements : ");
    scanf("%d", &n);

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

    if (A == NULL)
    {
        puts( "\n Memory was not allocated");
        exit(1);
    }
    puts( "\n Enter elements of array: ");
    for (i=0; i<n; i++)
    {
        scanf("%d", A+i );
    }
    puts ( "\n You entered following elements: ");

    for (i=0;i<n;i++)
    {
        printf(" %5d", *(A+i) );
    }
    printf ("\n");

    free(A);
    return 0;
}
```

**Ex.2: Appending new element at the end of
dynamic array by using function**

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

int* appendel(int *A, int *n, int apd);
int main( )
{
    int *A, n, i, apd, *p;
    clrscr( );
    printf("\n Enter number of elements : ");
    scanf("%d", &n);
    A = (int*) calloc( n, sizeof(*A) );
    if (A == NULL)
    {
        puts( "\n Memory was not allocated");
        return 1; }
    puts( "\n Enter elements of array: ");
    for (i=0; i<n; i++)
    { scanf("%d", &A[i] );}
    printf("Enter value of appending element: ");
    scanf("%d", &apd);
    p=appendel(A, &n, apd);
    if (p==NULL)
    { puts("\n Memory was not reallocated");
      return 1;}
    A= p;
    puts("Array with appended element:");
    for (i=0;i<n;i++)
    { printf(" %d\n", A[i] );}
    return 0;}

int* appendel(int *A, int *n, int apd)
{
    int i;
    int *p;
    p= (int *)realloc(A, (*n+1)*sizeof(*p) );
    if (p ==NULL) {return p;}
    p[*n] = apd;
    *n= *n+1;
    return p;
}
```

Ex.3: Inserting an element in dynamic array by using function

```
#include <stdio.h>
#include<stdlib.h>
int* insertel(int *A, int *n, int k, int ins);
int main( )
{ int *A, n, i, k, ins, *p;
  clrscr( );
  printf("\n Enter number of elements : ");
  scanf("%d", &n);
  A = (int*)malloc( n * sizeof(*A) );
  if (A == NULL)
  { puts( "\n Memory was not allocated");
    return 1; }
  puts ("\n Enter elements of array: ");
  for (i=0; i<n; i++)
  { scanf("%d", &A[i] );}
  printf("Enter position for insertion: ");
  scanf("%d", &k);
  k= k - 1;
  printf("Enter value of inserting element: ");
  scanf("%d", &ins);
  p=insertel(A, &n, k, ins);
  if (p==NULL)
  { puts("\n Memory was not reallocated");
    exit(1);}
  A= p;
  puts("Array with inserted element:");
  for (i=0;i<n;i++)
  { printf(" %d\n", A[i] );}

  return 0;
}

int* insertel(int *A, int *n, int k, int ins)
{ int i;
  int *p;
  p= (int *)realloc(A, (*n+1)*sizeof(*p) );
  if (p==NULL) {return p;}
  for(i=*n; i>k; i--)
  { p[i]= p[i-1]; }
  p[k]= ins;
  *n= *n+1;
  return p;
}
```

Ex.4: Deleting an element from dynamic array by using function

```
#include <stdio.h>
#include<stdlib.h>
int* delel(int *A, int *n, int k);
int main( )
{ int *A, n, i, k, *p;
  clrscr( );
  printf("\n Enter number of elements : ");
  scanf("%d", &n);
  A = (int*) malloc( n* sizeof(*A) );
  if (A == NULL)
  { puts( "\n Memory was not allocated");
    return 1; }
  puts ("\n Enter elements of array: ");
  for (i=0; i<n; i++)
  { scanf("%d", &A[i] );}
  printf("Enter position for deletion : ");
  scanf("%d", &k);
  k=k - 1;
  p=delel(A, &n, k);
  if (p==NULL)
  { puts("\n Memory was not reallocated");
    exit(1);}
  A= p;
  puts("Array with deleted element:");
  for (i=0;i<n;i++)
  { printf(" %d\n", A[i] );}

  return 0;
}

int* delel(int *A, int *n, int k)
{ int i;
  int *p;
  for(i=k; i<*n-1; i++)
  {A[i]= A[i+1];}
  p= (int *)realloc(A, (*n-1)*sizeof(*p) );
  if (p= NULL) {return p;}
  *n= *n-1;
  return p;
}
```