CSE115L – Computing Concepts Lab

Pointer declaration and referencing:

```
#include<stdio.h>
int main()
{
   int  var1;
   char var2;
   printf("Address of var1: %x\n",&var1);
   printf("Address of var2: %x\n",&var2);
   return 0;
}
```

```
#include <stdio.h>
int main ()
{
   int  *ptr = NULL;
   printf("The value of ptr is: %x\n",ptr);
   return 0;
}
```

```
#include<stdio.h>
int main()
{
   int  var=20;
   int *ip;
   ip=&var;
   printf("Address of var: %x\n",&var);
   printf("Address stored in ip: %x\n",ip);
   printf("Value of *ip : %d\n",*ip);

   return 0;
}
```

Accessing array elements using pointers:

```
#include <stdio.h>
int main ()
{
   int  arr[4]={2,5,1,6};
   int *ptr=arr;
   int i;
   for(i=0;i<4;i++)
   {
      printf("*ptr[%d]=%d\n",i,*(ptr+i));
   }
   return 0;
}
```

```
#include <stdio.h>
int main ()
{
   char str[]="hello";
   char *chptr;
   chptr=str;
   puts(str);
   puts(chptr);
   return 0;
}
```

```
#include <stdio.h>
int main ()
{
   char str[]="hello";
   char *chptr;
   chptr=str;
   int i=0;
   while(str[i]!='\0')
   {
      printf("%c",*chptr+i);
      i++;
   }
   return 0;
}
```

```
#include <stdio.h>
int main ()
{
   char str[]="hello";
   char *chptr;
   chptr=str;
   int i=0;
   while(str[i]!='\0')
   {
      printf("%c",*(chptr+i));
      i++;
   }
   return 0;
}
```

Dynamic memory allocation:

```
#include <stdio.h>
#include<stdlib.h>
int main ()
{
   int *data, i, n;
   scanf("%d",&n);
   data=(int*)malloc(n*sizeof(int));
   for(i=0;i<n;i++)
   {
     scanf("%d",data+i);
   }
   free(data);
   return 0;
}
```

Passing pointers as function arguments:

```c
#include <stdio.h>
void swap(int *p, int *q);
int main ()
{
   int a=2,b=3;
   swap(&a,&b);
   printf("a= %d b= %d",a,b);
   return 0;
}

void swap(int *p, int *q)
{
    int temp=*p;
    *p=*q;
    *q=temp;
}
```

```c
#include <stdio.h>
void print(char *s);
int main ()
{
   char str[]="simple";
   print(str);
   return 0;
}

void print(char *s)
{
    while(*s!='\0')
    {
      printf("%c",*s);
      s++;
    }
}
```

```c
#include <stdio.h>
void reverse(char *s);
int main ()
{
   char str[20];
   gets(str);
   reverse(str);
   puts(str);
   return 0;
}

void reverse(char *s)
{
   int i, len=0;
   for(i=0;s[i]!='\0';i++)
   {
     len++;
   }
   for(i=0;i<len/2;i++)
   {
     char temp=s[i];
     s[i]=s[len-i-1];
     s[len-i-1]=temp;

   }
}
```

Problems:

1. Implement the following function which finds the length of a string using pointer operation.

   `int length(char *s);`

2. Implement the following function which finds the largest element in an array using Dynamic Memory Allocation.

   `int max(int *p, int size);`

   Inside main, you have to create an array dynamically and pass the corresponding pointer and the array size to **`max`** function.