

Assignment: 1.1

Page 1

WAP to input n elements in an array and display the biggest and smallest element using function.

```
#include <stdio.h>
void element(int ar[], int n)
{
    for(i=0; i<n; i++)
    {
        int i, max = ar[0], min = ar[0];
        for(i=0; i<n; i++)
        {
            if(ar[i] > max)
                max = ar[i];
            if(ar[i] < min)
                min = ar[i];
        }
        printf(" Biggest element = %d \n", max);
        printf(" smallest element = %d \n", min);
    }
}
```

```
int main()
{
    int n, i, ar[20];
    printf(" Enter size of array: ");
    scanf("%d", &n);
    printf(" Enter elements in the array: ");
    for(i=0; i<n; i++)
    {
        scanf("%d", &ar[i]);
    }
    element(ar, n);
    return 0;
}
```

Assignment: 1.2

page?

WAP to dynamically allocate memory for n elements in an array and display 2nd biggest and 2nd smallest element

```
#include <stdio.h>
#include <stdlib.h>
void sec(int *, int);
void main()
{
    int i, *p, n;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    p = (int *) malloc(n * sizeof(int));
    printf("Enter elements in array: ");
    for (i=0; i<n; i++)
    {
        scanf("%d", &p[i]);
    }
    sec(p, n);
}
```

```
void sec(int *p, int n)
```

```
{
    int i, j, temp;
    for (i=1; i<n; i++)
    {
        for (j=0; j<n-i; j++)
        {
            if (p[j] > p[j+1])
            {
                temp = p[j];
                p[j] = p[j+1];
                p[j+1] = temp;
            }
        }
    }
}
```

```
printf("2nd biggest number = %d \n", p[n-2]);
printf("2nd smallest number = %d \n", p[1]);
```

Assignment : 1.3

#in WAP to create an array with n number of random numbers and arrange them in descending order.

```
#include <stdio.h>
#include <stdlib.h>
void arrange (int *p, int n)
{
    void main()
    {
        int n, *p, i;
        printf("Enter number of elements: ");
        scanf("%d", &n);
        p=(int *) malloc (n * sizeof(int));
        for(i=0; i<n; i++)
        {
            p[i] = random() % 10 + i;
        }
        printf("elements of array: ");
        for(i=0; i<n; i++)
        {
            printf("%d ", p[i]);
        }
        arrange(p, n);
    }
}
```

```
void arrange (int *p, int n)
{
    int i, j, temp;
    for(i=1; i<n; i++)
    {
        for(j=0; j<n-i; j++)
        {
            if(p[j] < p[j+1])
            {
                temp = p[j];
                p[j] = p[j+1];
                p[j+1] = temp;
            }
        }
    }
}
```

```
printf("Elements in descending order: ");
```

```
for(i=0; i<n; i++)
```

```
{
```

```
    printf("%d ", p[i]);
```

```
}
```

```
}
```

Assignment 1.4

Page 2

WAP to insert an element at a particular position and delete an element from a particular position using function.

```
#include <stdio.h>
void insert(int[], int *, int, int);
void delete(int[], int *, int, int);
int void display(int[], int);

int main()
{
    int ar[30], n, i;
    scanf("%d", &n);
    for(i=0; i<n; i++)
    {
        scanf("%d", &ar[i]);
    }

    int p1, num1, p2, num2;
    printf("Enter element to insert and pos: ");
    scanf("%d %d", &num1, &p1);
    display(ar, n);
    printf("Enter element to display and pos: ");
    scanf("%d %d", &num2, &p2);
    display(ar, n);

    void insert(int ar[], int *n, int p1, int num1)
    {
        int i;
        for(i = *n-1; i >= p1; i--)
        {
            ar[i+1] = ar[i];
        }
        ar[p1] = num1;
        *n = *n + 1;
    }

    void delete(int ar[], int *n, int p1, int num1)
    {
        int i;
        for(i = p1; i < *n-1; i++)
        {
            ar[i] = ar[i+1];
        }
        *n = *n - 1;
    }

    if(p1 < 0 || p1 > n)
    {
        printf("Invalid position");
    }
    else
    {
        if(num1 < 0)
        {
            printf("Invalid number");
        }
        else
        {
            insert(ar, &n, p1, num1);
            display(ar, n);
        }
    }
}
```

```
void delete (int ar[], int *n, int p2, int num2)
{
    int i;
    for(i=p2; i<*n-1; i++)
    {
        ar[i] = ar[i+1];
    }
    *n = *n - 1;
}
```

```
void display (int ar[], int n)
```

```
{
    int i;
    printf("Array elements : ");
    for(i=0; i<n; i++)
        printf("%d", ar[i]);
```

```
}
```

Write a menu driven program to

- (1) generate random numbers
- (2) Display the elements
- (3) Insert new element at beginning of array
- (4) Insert element at ~~in~~ end
- (5) Insert element at middle
- (6) Exit

```
#include <stdio.h>
#include <stdlib.h>
void ran(mt *p, int n)
{
    int i;
    for(i=0; i<n; i++)
        p[i] = random() % 10 + i;
}
```

```
void display(int *p, int n)
{
    int i;
    for(i=0; i<n; i++)
        printf("%d ", p[i]);
```

```
void insertf(int *p, int *n, int num)
{
    int i;
    for(i=*n-1; i>=0; i--)
    {
        p[i+1] = p[i];
    }
    p[0] = num;
    *n = *n + 1;
```

```

void inserte(int p[], int *n, int num)
{
    p[*n] = num;
    *n = *n + 1;
}

void insertm(int p[], int *n, int num)
{
    int i;
    for(i = *n - 1; i >= *n / 2; i--)
    {
        p[i + 1] = p[i];
    }
    p[n / 2 - 1] = num;
    *n = *n + 1;
}

int main()
{
    int *p, n, i, ch, ans;
    printf("enter number of elements: ");
    scanf("%d", &n);
    p = (int *)malloc(n * sizeof(int));

    do
    {
        printf("press 1 to create a array\n");
        printf("press 2 to display \n");
        printf("press 3 to insert at beg \n");
        printf("press 4 to insert at end \n");
        printf("press 5 to insert at middle \n");
        printf("press 6 to exit: \n");
        printf("Enter your choice: ");
        scanf("%d", &ch);

        int num;
        switch(ch)
    }

```

```
{  
    case 1: ran(p,n);  
        break;  
    case 2: display(p,n);  
        break;  
    case 3: printf("Enter num to insert : ");  
        scanf("%d", &num);  
        insertf(p, &n, num);  
        break;  
    case 4: printf("Enter num to insert at end : ");  
        scanf("%d", &num);  
        insertl(p, &n, num);  
        break;  
    case 5: printf("Enter num to insert at middle : ");  
        scanf("%d", &num);  
        insertm(p, &n, num);  
        break;  
    case 6: exit(0);  
    default: printf("Invalid choice: ");
```

} printf("Do you want to continue, press 0 ");
scanf("%d", &ans);
} while(ans == 0);

}

Assignment 1.6

Page 10

WAP to arrange the first half of the array in ascending and 2nd half in descending order.

```
#include <stdio.h>
void swap(int *a1, int *a2)
{
    int temp;
    temp = *a1;
    *a1 = *a2;
    *a2 = temp;
}

void sort(int ar[], int n)
{
    int i, j;
    for(i=1; i<n/2; i++)
    {
        for(j=0; j<n/2 - i; j++)
        {
            if(ar[j] > ar[j+1])
                swap(&ar[j], &ar[j+1]);
        }
    }

    for(i=1; i<n-n/2; i++)
    {
        for(j=n/2; j<n-i; j++)
        {
            if(ar[j] < ar[j+1])
                swap(&ar[j], &ar[j+1]);
        }
    }
}
```

```
void display(int ar[], int n)
{
    printf("rearranged array: ");
    for(int i=0; i<n; i++)
        printf("%d ", ar[i]);
}

int main()
{
    int i, n, ar[30];
    printf("Enter size of array: ");
    scanf("%d", &n);
    printf("Enter elements: ");
    for(i=0; i<n; i++)
        scanf("%d", &ar[i]);
    sort(ar, n);
    display(ar, n);
}
```

2

Assignment: 1.7

page 12

WAP to delete all duplicate elements from the array

```
#include <stdio.h>
void delete (int ar[], int *n, int pos)
{
    int i;
    for(i = pos; i < n - 1; i++)
    {
        ar[i] = ar[i + 1];
    }
    *n = *n - 1;
}

int main()
{
    int n, ar[30], i;
    printf("Enter number of elements:");
    scanf("%d", &n);
    for(i = 0; i < n - 1; i++)
    {
        for(j = i + 1; j < n; j++)
        {
            if(ar[i] == ar[j])
                delete(ar, &n, j);
        }
    }
    printf("array elements:");
    for(i = 0; i < n; i++)
        printf("%d", ar[i]);
}
```

3

Assignment: 1-8

Using structure input and display the details of a student

```
#include <stdio.h>
struct stu
{
    char nm[20], br[10];
    int roll, cgpa;
}s;
int main()
{
    printf("enter details\n");
    printf("enter name:");
    gets(s.nm);
    printf("enter roll:");
    scanf("%d", &s.roll);
    while(getchar() != '\n');
    printf("enter branch:");
    gets(s.br);
    printf("enter cgpa:");
    scanf("%d", &s.cgpa);

    printf("student details:\n");
    printf("name: %s\n", s.nm);
    printf("roll no: %d\n", s.roll);
    printf("branch: %s\n", s.br);
    printf("cgpa: %d\n", s.cgpa);
```

Assignment 1.9

Page 14

Input and display (in tabular form) details of 'n' employee.

```
#include <stdio.h>
struct emp
{
    char nm[20], desg[20], dept[20];
    int basic, empid;
};

void input(struct emp e[], int n)
{
    int i;
    for(i=0; i<n; i++)
    {
        printf("Enter details of employee %d\n", i+1);
        while(getchar() != '\n');
        printf("Enter name: ");
        gets(e[i].nm);
        printf("Enter designation: ");
        gets(e[i].desg);
        printf("Enter department: ");
        gets(e[i].dept);
        printf("Enter employee id: ");
        scanf("%d", &e[i].empid);
        printf("Enter basic: ");
        scanf("%d", &e[i].basic);
    }
}
```

3

```

void display (struct emp e[], int n)
{
    int i;
    printf ("Employee details:\n");
    printf ("% -5d % -15d % -15d % -15d % -15d", "st", "name",
            "empid", "designation", "department", "basic");
    for (i=0; i < n; i++)
    {
        printf ("\n");
        printf ("% -5d % -15d % -15d % -15d % -15d", i+1,
                e[i].name, e[i].empid, e[i].desg, e[i].dept, e[i].basic);
    }
}

```

3

)

int main()

```

{
    int i, n;
    scanf ("%d", &n);
    struct emp e[n];
    input (e, n);
    display (e, n);
}

```

3

Input details of 'n' no. of employees and display in tabular form and also calculate TA, DA, HRA and Gross.

```
#include <stdio.h>
struct emp
{
    char nm[30], desg[20], dept[20];
    int basic, empid;
    float gross;
};

void input(struct emp e[], int n)
{
    int i;
    for(i=0; i<n; i++)
    {
        printf("enter details of employees: %d \n", i+1);
        while(getchar() != '\n');
        printf("enter name : ");
        gets(e[i].nm);
        printf("enter designation: ");
        gets(e[i].desg);
        printf("enter employee id : ");
        scanf("%d", &e[i].empid);
        printf("enter department: ");
        gets(e[i].dept);
        printf("enter basic salary : ");
        scanf("%d", &e[i].basic);
    }
}
```

3

```

void display (struct emp e[], int n)
{
    int i;
    printf(" Employee details :\n");
    printf("%-5s %-15s %-15s %-15s %-15s %-15s %-15s\n",
           "sl.no", "name", "empid", "designation", "department",
           "basic", "gross");
    for(i=0; i<n; i++)
    {
        printf("\n");
        printf("%-5d %-15s %-15d %-15s %-15s %-15s %-15s\n",
               i+1, e[i].nm, e[i].empid, e[i].desg, e[i].dept, e[i].basic,
               e[i].gross);
    }
}

void calculate (struct emp e[], int n)
{
    floathra, da, ta;
    for(int i=0; i<n; i++)
    {
        hra = 0.06 * e[i].basic;
        ta = 0.07 * e[i].basic;
        da = 0.08 * e[i].basic;
        e[i].gross = e[i].basic + hra + da + ta;
    }
}

int main()
{
    int i, n;
    printf(" Enter no. of employees : ");
    scanf("%d", &n);
    struct emp e[n];
    input(e, n);
    calculate(e, n);
    display(e, n);
}

```