DS Lab Assignment - 4

14/09/2021

Q.1. Create a structure for a company who manufactures mobile phones. Create a repository of information which consists of information which consists of the following data:

a. prod_name b. prod_code c. prod_price d. prod_screensize e. prod_colour.

Now, accept data for 5 different products and display them in a well-formatted manner. Use structure and arrays.

Ans:- Source Code :-

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

struct product {
    char prod_name[50];
    char prod_code[20];
    int product_price;
    float prod_screensize;
    char prod_color[20];
} mobile[5];

void main (){
    struct product *ptr;
    ptr = mobile;

    for( int i = 0; i< 5; i++) {
        printf("Enter details of mobile %d \n", i+1);
        printf("Product Name : ");
        gets ((ptr+i) -> prod_name);
        printf("\t Product Code : ");
        gets ((ptr+i) -> prod_code);
        printf("\t Product Price : ");
        scanf("%d", &(ptr+i) -> product_price);
```

```
        printf(" Product Screensize :  ");
        scanf("%f", &(ptr+i)->prod_screensize);
        printf(" Product Color :  ");
        fflush(stdin);
        gets((ptr+i)->prod_color);
}

    printf("Details of Products :-\n");
    for(int i=0 ; i<5 ; i++) {
    printf("\n Details of mobile %d are\n", i+1);
    printf(" Product Name : %s \n", (ptr+i)->prod_name);
    printf(" Product Code : %s \n", (ptr+i)->prod_code);
    printf(" Product Price : %d \n", (ptr+i)->product_price);
    printf(" Product Screensize: %.2f \n", (ptr+i)->prod_screensize);
    printf("Product Color : %s \n", (ptr+i)->prod_color);

    }

}
```

Output :-

```
Enter details of mobile 1
Product Name : iPhone 11
Product Code : X100259
Product Price : 70000
Product Screensize : 6.00
Product Colour : Red.
Enter details of mobile 2
Product Name : Vivo V9
Product Code : XYN0034
Product Price : 18000
Product Screensize : 6.5
Product Color : Cyan.
```

Enter details of mobile 3

Product Name : Samsung A30

Product Code : A399762

Product Price : 20000

Product Screensize : 6.4

Product Color : White

Enter details of mobile 4.

Product Name : Realme X3

Product Code : X37A06

Product Price : 30000

Product Screensize : 7

Product Colour : Magenta

Enter details of mobile 5

Product Name : Redmi 9

Product Code : CH0097

Product Price : 40000

Product Screensize : 6.3

Product Color : Yellow.

Details of Products :-

Details of mobile 1 are

Product Name : iPhone 11

Product Code : X100259

Product Price : 70000

Product Screensize : 6.00

Product Color : Red.

Details of mobile 2 are

Product Name : Vivo V9

Product Code : XYA0054

Product Price : 18000

Product Screensize : 6.50

Product Color : Cyan.

Details of mobile 3 are.

Product Name : Samsung A30

Product Code : AB99762

Product Price : 20000

Product Screensize : 6.40

Product Color : White.

Details of mobile 4 are

Product Name : Realme X3

Product Code : X37A06

Product Price : 30000

Product Screensize : 7.00

Product Color : Magenta.

Details of mobile 5 are

Product Name : Redmi 9

Product Code : CH0097

Product Price : 40000

Product Screensize : 6.30

Product Colour : Yellow.

Q.2. Implement Dynamic Memory Allocation ~~and dis~~ to demonstrate the usage and implementation of the Following :- a. malloc() b. calloc() c. free().

Source Code :

```
#include <stdio.h>
#include <stdlib.h>
void display (int *ptr, int n);
void input (int * ptr , int n);
void main ()
    int * arr, n;
    printf (" Enter size of array : ");
    scanf ("%d", &n);
    arr = (int *) malloc (sizeof (int) * n);
    input (arr, n);
    display (arr, n);
    free (arr);
    display (arr, n);
    arr = (int *) calloc (n, sizeof (int));
    display (arr, n);
    free (arr);
    display (arr, n);
}
```

```
void input (int *ptr, int n) {
    printf ("\nEnter elements into the array : \n");
    for( int i=0 ; i <n ; i++ ){
       scanf ("%d" , ptr+i);
    }
    printF ("\n");
}

void display (int * ptr, int n) {
    printf (" Elements of array are : \n");
    for (int i=0 ; i<n; i++)
       printF (" %d ", *(ptr+i));
    printF ("\n");
}
```

Output :-

Enter size of array : 4

Enter elements into the array :
2 3 5 1

Elements of array are :
2 3 5 1

Elements of array are :
498321 -12354 98706 0

Elements of array are :
0 0 0 0

Element of array are :

498821 -12354 98706 0

Q.3. Implement DMA to accept N integer Data and find sum of composite integers only. Also display the memory locations along the data.

Ans :-

Source code :

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int sum = 0;
int prime (int num) {
    int fact = 0, temp = 1;
    while (num + 1 > temp) {
        if (num % temp == 0)
            fact;
        temp;
    }
    return fact;
}

void main() {
    int *arr, N;
    printf ("Enter the size of array : ");
    scanf ("%d", &N);
    arr = (int *) malloc (sizeof(int) * N);

    printf (" Enter integer data : \n");
    for (int = 0; i<N; i++) {
        printf ("arr[%d] : ", i);
        scanf (" %d", &arr[i]);

    }
```

```
printf("\nInput Array : \n");
for (int i=0 ; i<N ; i++) {
    if (prime (arr[i]) != 2)
        sum += arr[i];
printf(" value: %d :: Address : %x\n",
*(arr+i), arr+i);
}
printf("\n Sum of composite integers : %d", sum);
}
```

Output :-

```
Enter size of array : 5
Enter integer data :
arr [0] : 0
arr [1] : -2
arr [2] : 3
arr [3] : 5
arr [4] : 6

Input array :
Value : 0 :: Address : 9779eac0
Value : -2 :: Address : 9779eac4
Value : 3 :: Address : 9779eac8
Value : 5 :: Address : 9779eacc
Value : 6 :: Address : 9779ad0

Sum of composite integers : 4
```

0.4. Implement DMA and create a memory location of size N. Accept the data by passing the base address of the allocation to a function and display the memory locations inside the function.

Source code:

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

void myfun (int * ptr, int len) {
    printf("\n Enter your data : \n");
    for (int i = 0; i <len; i++) {
        printf(" arr [%d] : ", i);
        scanf ("%d", ptr +i);
    }

    printf("\n The input array is : \n");
    for (int i=0 ; i <len ; i++) {
        printf ("value : %d :: Address : %x\n",
        * (ptr+i), (ptr+i));
    }
}

void main (){
    int * arr, N;
    printf (" Enter size of array: ");
    scanf ("%d", &N);
    arr = (int *)malloc (sizeof(int) * N);
    myfun (arr, N);
    Free (arr);
}
```

Output :-

```
Enter size of array : 4

Enter your data :
arr[0] : 3
arr[1] : 4
arr[2] : -6
arr[3] : 9

The input array is :
Value : 3   :: Address : 9215dac0
Value : 9   :: Address : 9215dac4
Value : -6  :: Address : 9215dac8
Value : 5   :: Address : 9215dacc
```