Master arrays

More Functions on arrays:

01- Find sum negatives and positive integers
02- Max & Min of given numbers
03- Find two first maximum numbers
04- Separate Even, odd numbers
05- Insert an element at a specified position
06- Delete a specified element
07- Remove repeated elements
08- Merge two arrays in sorted order
09- Union & intersection of the given array
10- Sum of two numbers equal to X, X integer given
11- Average of numbers at even position
12- Array elements in reverse order using swapping
13- Search for a specific element
14- Binary search

More Functions on arrays:

01- Find sum of negative and positive integers

```c
// 01- Find sum of negative and positive integers
#include <stdio.h>

int main()
{
     int array[] = {-22, -17, 2, 5, 3, 4, 6, 8, 7, 1, 9};
     int size = sizeof(array) / sizeof(array[0]);
     int i, negs = 0, pos = 0;
     for (i = 0; i < size; i++)
     {
          if (array[i] < 0)
               negs += array[i];
          if (array[i] > 0)
               pos += array[i];
     }

     printf("Sum of negatives: %d\n", negs);
     printf("Sum of positives: %d\n", pos);

     return 0;
}
```

02- Max & Min of given numbers

```c
// 02- Max & Min of given numbers
#include <stdio.h>

int main()
{
     int array[] = {-22, -17, 2, 5, 3, 4, 6, 8, 7, 1, 9, 13};
     int size = sizeof(array) / sizeof(array[0]);
     int i;
     int min = array[0];
     int max = array[0];

     for (i = 0; i < size; i++)
     {
          if (min > array[i])
               min = array[i];
          if (max < array[i])
               max = array[i];
     }

     printf("Minimum: %d\n", min);
     printf("Maximum: %d\n", max);

     return 0;
}
```

03- Find two first maximum numbers

```c
// 03-    Find two first maximum numbers
#include <stdio.h>
int main()
{
     int i, size, max, max2, position = 0;
     int a[] = {-22, 2, 5, 3, 1, 9, 13};
     size = sizeof(a) / sizeof(a[0]);
     int temp[size - 1];
     max = a[0];

     // find first max number
     for (i = 0; i < size; i++)
     {
          if (a[i] > max)
          {
               max = a[i];
               position = i;
          }
     }

     // deleting first maximum number in array
     for (i = 0; i < size - 1; i++)
     {
          if (i < position)
               temp[i] = a[i];
          if (i >= position)
               temp[i] = a[i + 1];
     }

     max2 = temp[0];
     // finding second max in the remaining elements
     for (i = 0; i < size - 1; i++)
     {
          if (temp[i] > max2)
               max2 = temp[i];
     }
     printf("The fisrt largest number is %d\n", max);
     printf("The second largest number is %d\n", max2);
}
```

04- Separate Even, odd numbers

```c
// 04- Separate Even, odd numbers
#include <stdio.h>
int main()
{
     int i, size;
     int a[] = {-22, 2, 5, 3, 1, 9, 13, 77, 88, 99, 16, 14, 24};
     size = sizeof(a) / sizeof(a[0]);

     printf("Even numbers are:\n");
     for (i = 0; i < size; i++)
          if (a[i] >= 0)
               if (a[i] % 2 == 0)
                    printf("%d\t", a[i]);

     printf("\nOdd numbers are:\n");
     for (i = 0; i < size; i++)
          if (a[i] >= 0)
               if (a[i] % 2 == 1)
                    printf("%d\t", a[i]);
}
```

05- Insert an element at a specified position

```c
// 05- Insert an element at a specified position
#include <stdio.h>
int main()
{
     int i, size, position, insitem;
     int a[] = {-22, 2, 5, 3, 1, 9, 13};
     size = sizeof(a) / sizeof(a[0]);
     int temp[size + 1];

     printf("Enter element to be inserted: ");
     scanf("%d", &insitem);
     printf("Enter position for an element to be inserted: ");
     scanf("%d", &position);
     position = position - 1;
     for (i = 0; i <= size; i++)
     {
          if (i < position)
               temp[i] = a[i];
          if (i > position)
               temp[i] = a[i - 1];
          if (i == position)
               temp[i] = insitem;
     }
     printf("Array after inserting %d\n", insitem);
     for (i = 0; i <= size; i++)
          printf("%d\t", temp[i]);
}
```

06- Delete a specified element

```c
// 06- Delete a specified element
#include <stdio.h>
int main(){
     int i, size, position, delitem, flag = 0;
     int a[] = {-22, 2, 5, 3, 1, 9, 13};
     size = sizeof(a) / sizeof(a[0]);
     int temp[size - 1];
     for (i = 0; i < size; i++)
          printf("%d\t", a[i]);
     printf("\nEnter element to be deleted: ");
     scanf("%d", &delitem);
     for (i = 0; i < size; i++) // find position of a number
     {
          if (a[i] == delitem)
          {
               position = i;
               flag = 1;
          }
     }
     if (flag == 1) {
          for (i = 0; i < size - 1; i++) // deleting number in array
          {
               if (i < position)
                    temp[i] = a[i];
               if (i >= position)
                    temp[i] = a[i + 1];
          }
          printf("Array after deleting %d\n", delitem);
          for (i = 0; i < size - 1; i++)
               printf("%d\t", temp[i]);
     }
     else
          printf("Number not found in array\n");
}
```

07- Remove repeated elements

```c
// 07- Remove repeated elements
#include <stdio.h>
void removerep(int position, int a[], int size);
int main() {
     int i, j, size, count = 0, k, position;
     int a[] = {-22, 2, 5, 3, 1, 9, 13};
     size = sizeof(a) / sizeof(a[0]);
     int *temp = a;
     for (i = 0; i < size; i++)
     {
          for (j = i + 1; j < size;)
          {
               if (a[i] == a[j])
               {
                    removerep(j, a, size);
                    size--;
               }
               else
                    j++;
          }
     }
     printf("After removing repeated elements\n");
     for (i = 0; i < size; i++)
          printf("%d\n", a[i]);
}
void removerep(int position, int a[], int size)
{
     int i;
     for (i = 0; i < size; i++) // deleting repeated number in array
     {
          if (i < position)
               a[i] = a[i];
          if (i >= position)
               a[i] = a[i + 1];
     }
}
```

/// Method II :

```c
// 07- Remove repeated elements
#include <stdio.h>
int main()
{
     int i, j, k, size, count = 0, position;
     int a[] = {-22, 2, 5, 3, 1, 9, 13};
     size = sizeof(a) / sizeof(a[0]);
     int *temp = a;
     for (i = 0; i < size; i++)
     {
          for (j = i + 1; j < size;)
          {
               if (a[i] == a[j])
               {
                    for (k = j; k < size; k++)
                         a[k] = a[k + 1];
                    size--;
               }
               else
                    j++;
          }
     }
     printf("After removing repeated elements\n");
     for (i = 0; i < size; i++)
          printf("%d\n", a[i]);
}
```

08- Merge two arrays in sorted order

```c
// 08- Merge two arrays in sorted order

#include <stdio.h>
int main()
{
     int i, size1, size2, j = 0, temp;
     int a1[] = {1, 2, 3, 4};
     size1 = sizeof(a1) / sizeof(a1[0]);

     int a2[] = {5, 6, 7, 8};
     size2 = sizeof(a2) / sizeof(a2[0]);
     int a3[size1 + size2];

     // merging
     for (i = 0; i < size1; i++)
     {
          a3[j] = a1[i];
          j++;
     }
     for (i = 0; i < size2; i++)
     {
          a3[j] = a2[i];
          j++;
     }
     printf("\nArray after merging\n");
     for (i = 0; i < size1 + size2; i++)
          printf("%d\t", a3[i]);

     return 0;
}
```

09- Union & intersection of the given array

```c
// 09-    Union & intersection of the given array
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
void display(int array[], int size);
void unions(int array1[], int array2[], int size1, int size2);
void intersection(int array1[], int array2[], int size1, int size2);

int main()
{
     int size1, size2;
     int array1[] = {1, 2, 3, 4, 5};
     int array2[] = {1, 2, 3, 6, 7, 8, 9, 10};
     size1 = sizeof(array1) / sizeof(array1[0]);
     size2 = sizeof(array2) / sizeof(array2[0]);
     printf("array1:\t");
     display(array1, size1);
     printf("array2:\t");
     display(array2, size2);
     intersection(array1, array2, size1, size2);
     printf("\n");
     unions(array1, array2, size1, size2);
     return 0;
}
void display(int array[], int size)
{
     for (int i = 0; i < size; i++)
          printf("%d, ", array[i]);
     printf("\n");
}
```

```c
void intersection(int array1[], int array2[], int size1, int size2)
{
     printf("Inters:\t");
     for (int i = 0; i < size1; i++)
          for (int j = 0; j < size2; j++)
               if (array1[i] == array2[j])
                    printf("%d, ", array1[i]);
}

void unions(int array1[], int array2[], int size1, int size2)
{
     int i, j, k, size, count = 0;
     size = size1 + size2;
     int array[size];
     // merge
     for (i = 0; i < size1; i++)
     {
          array[count] = array1[i];
          count++;
     }
     for (i = 0; i < size2; i++)
     {
          array[count] = array2[i];
          count++;
     }

     // remove repeated elements
     for (i = 0; i < size - 1; i++)
     {
          for (j = i + 1; j < size;)
          {
               if (array[i] == array[j])
               {
                    for (k = j; k < size; k++)
                    {
                         if (k < j)
                              array[k] = array[k];
                         if (k >= j)
                              array[k] = array[k + 1];
                    }
                    size--;
               }
               else
                    j++;
          }
     }

     printf("Unions:\t");
     display(array, size);
}
```

10- Sum of two numbers equal to X, X integer given

```c
// 10-    Sum of two numbers equal to X, X integer given
#include <stdio.h>
#include <stdlib.h>

int main()
{
     int x, size;
     int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
     size = sizeof(array) / sizeof(array[0]);
     printf("Enter number: ");
     scanf("%d", &x);

     for (int i = 0; i < size - 1; i++)
          for (int j = i + 1; j < size; j++)
               if (array[i] + array[j] == x)
                    printf("%d + %d = %d\n", array[i], array[j], x);

     return 0;
}
```

11- Average of numbers at even position

```c
// 11- Average of numbers at even position
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

int main()
{
     int i, size, sum = 0;
     int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
     size = sizeof(array) / sizeof(array[0]);
     for (i = 0; i < size; i++)
     {
          if ((i + 1) % 2 == 0)
               sum += array[i];
     }
     printf("Sum = %d\n", sum);
     printf("Average = %d\n", sum / 2);

     return 0;
}
```

12- Array elements in reverse order using swapping

```c
// 12-    Array elements in reverse order using swapping
#include <stdio.h>
#include <stdlib.h>

int main()
{
     int i, k, size, temp;
     int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
     size = sizeof(array) / sizeof(array[0]);

     printf("Before reverse: ");
     for (i = 0; i < size; i++)
          printf("%d, ", array[i]);

     for (k = size; k >= 0; k--)
     {
          for (i = 0; i < k - 1; i++)
          {
               temp = array[i];
               array[i] = array[i + 1];
               array[i + 1] = temp;
          }
     }
     printf("\nAfter reverse : ");
     for (i = 0; i < size; i++)
          printf("%d, ", array[i]);

     return 0;
}
```

13- Search for a specific element

```c
// 13-    Search for a specific element
#include <stdio.h>
#include <stdlib.h>

int main()
{
     int i, x, size, found = 0;
     int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
     size = sizeof(array) / sizeof(array[0]);

     for (i = 0; i < size; i++)
          printf("%d, ", array[i]);

     printf("\nFind: ");
     scanf("%d", &x);

     for (i = 0; i < size; i++)
          if (x == array[i])
               printf("%d found at position %d\n", x, i);

     return 0;
}
```

Created by Muhammed El Idrissi Laoukili

لااله الا الله