1)

A) declare and initialize an array of twenty 32-bit signed integers.

B) define a function named "even_sum", that takes two parameters.
The first parameter should be a pointer to a 32-bit integer to pass in the array and the second should be a 32-bit integer that you will use to pass the length of the previously declared array. The function should return the sum of the array elements with even indices.

Answer 1.A)

```
int main() {

    //declaring
    int arr[20]; //{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20};
    int size; //{20}
    int *p;
    printf("Enter size of the array: \n");
    scanf("%d",&size);
    //initializing
    printf("Enter the elements of the array: \n");
    for(int i=0; i<size; i++)
    {
        scanf("%d",&arr[i]);
    }
    // assign the address of int
    p=&arr[0];
}
```

OR

```
int main() {

//declaring and initializing
int arr[20] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20};
int size = 20;
int *p;
}
```

Answer 1.B)

```
int even_sum(int *p, int size)
{
    int sum=0;
    for(int i=0; i<size; i++)
  {
      if(i%2==0)
      {sum+=p[i];}
      else
      {p[i]=0;}
  }
  return sum;
}
int result=even_sum(p, size); //inside int main()
```

OUTPUT -

It returns 100 as the sum

2) Modify the previous function to also set all elements at odd indices to zero.
Answer 2)

```
void zero_structs (struct Student *ptr, int size)
{
        for(int i=0; i<size; i++)
        {
        ptr[i].roll_no = 0;
        ptr[i].reg_no = 0;
        ptr[i].percentage = 0;
        ptr[i].net_id = 0;
        }
}

zero_structs (classs, size_class); //inside int main()
```

OUTPUT – all the even numbers are now set 0.

3) Define a struct with 4 members, each member should be an unsigned int. Declare an array with 5 of the previously defined structs.

Answer 3)

```
struct Student {
    unsigned int   net_id;
    unsigned int   reg_no;
    unsigned int   roll_no;
    unsigned int   percentage;
};



struct Student classs[5]; //inside int main()
```

4) define a function named "zero_structs", that takes 2 parameters. The first parameter should be a pointer to a struct of the previously defined type to pass in the array of structs, and the second should be a 32-bit integer that you will use to pass the length of the array. The function should set all members of the structs in the passed-in array to zero.

Answer 4)

```
void zero_structs (struct Student *ptr, int size)
{
        for(int i=0; i<size; i++)
        {
        ptr[i].roll_no = 0;
        ptr[i].reg_no = 0;
        ptr[i].percentage = 0;
        ptr[i].net_id = 0;
        }
}

zero_structs (classs, size_class); //inside int main()
```

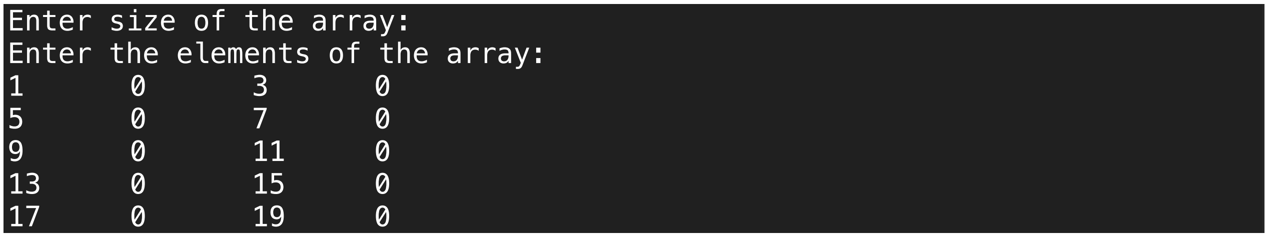
5) Define a function named "fill_structs", which takes 3 parameters. The first parameter should be a pointer to a struct of the previously defined type to pass in the array of structs, the second should be a pointer to a 32-bit integer to pass in the array from part one, and the third should be a 32-bit integer that you will use to pass the length of the integer array. The function should set the members of the structs in the passed-in array to the elements of the 32-bit integer array that you pass in (i.e. the first struct's members should have the first 4 values of the integer array, the second struct should have the next 4 values, etc.)

Answer 5)

```
void fill_structs(struct Student *ptr,int *p, int size)
{
    for(int i=0; i<size; i=i+4)
        {
        int j=i/4;
        ptr[j].net_id = p[i];
        ptr[j].reg_no = p[i+1];
        ptr[j].roll_no = p[i+2];
        ptr[j].percentage = p[i+3];
        }
}


//inside int main() with additional print statements
fill_structs(classs, p, size);
    for(int i=0; i<size_class; i++)
    {
        printf ("%d\t", classs[i].net_id);
        printf ("%d\t", classs[i].reg_no);
        printf ("%d\t", classs[i].roll_no);
        printf ("%d\n", classs[i].percentage);
    }
```

OUTPUT –



```
Enter size of the array:
Enter the elements of the array:
1       0       3       0
5       0       7       0
9       0       11      0
13      0       15      0
17      0       19      0
```