```
#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/wait.h>

int main()
{
	pid_t pid1,n,pid;
	n = fork();
	if( n == 0 )
	{
		printf("hello this is a child process.\n");
		pid = getpid();
		printf("process id = %d\n",pid);
	}
	else if( n >=1 )
	{
		wait(NULL);
		printf("hello this is a parent process.\n");
		pid = getpid();
		printf("process id = %d\n",pid);
	}
	else
	{
		printf("Process creation failed\n");
	}
	return 0;
}
```

output:

```
hello this is a child process.
process id = 7051
hello this is a parent process.
process id = 7050
```

SLEEP:>>

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

int main() {
   printf("Sleep for 5 seconds...\n");
```

```
    sleep(5); // Sleep for 5 seconds
    printf("Awake now!\n");

    return 0;
}
```

Background process:

```
 ./a.out &
[2] 1084
```

2A

```
#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/wait.h>

int main() {
    pid_t pid;
    pid = fork();

    if (pid == 0) {
        printf("this is a child process\n");
    } else {
        // Wait for the first child to finish in the parent process
        wait(NULL);
        printf("this is a parent process\n");
    }
    return 0;
}
```

/array////////

program>>>

```
#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
```

```
#include<sys/wait.h>

int main() {
    int size;
    printf("Enter the size of the array: ");
    scanf("%d", &size);

    int arr[size];

    printf("Enter array values:\n");
    for (int i = 0; i < size; i++) {
        printf("Enter the value for element %d: ", i);
        scanf("%d", &arr[i]);
    }

    pid_t pid;
    pid = fork();

    if (pid == 0) {
        printf("Child Process - Sorting Array: ");
        // Use your sorting algorithm here (e.g., bubble sort, quicksort, etc.)
        // Just for demonstration, using a simple bubble sort
        for (int i = 0; i < size - 1; i++) {
            for (int j = 0; j < size - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {
                    // Swap elements if they are in the wrong order
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }

        printf("Sorted Array in Child Process: ");
        for (int i = 0; i < size; i++) {
            printf("%d ", arr[i]);
        }
        printf("\n");
    } else {
        wait(NULL);
        printf("Array in Parent Process: ");
        for (int i = 0; i < size; i++) {
            printf("%d ", arr[i]);
        }
```

```
        printf("\n");
     printf("ran once\n");
    }

    return 0;
}
```

```
Enter the size of the array: 4
Enter array values:
Enter the value for element 0: 1
Enter the value for element 1: 1
Enter the value for element 2: 2
Enter the value for element 3: 3
Child Process - Sorting Array: Sorted Array in Child Process: 1 1 2 3
Array in Parent Process: 1 1 2 3
ran once
```