1.give *sleep()* a random number between 1-10
Use SIGALRM to making parent process kills itself, and once *getppid()==1 the* child process kills itself.

2.
sort system call:

```
sprintf(cmd,"sort -n -o %s %s",dirname,dirname);
system(cmd);
```

Parent process store the answer from child processes to a struct.

After all the files are sorted, the main process closes the request_pipe [1], so the children stop sending requests and closing all pipe ends.

```
    if(read(answer_pipe[0], buffer3, BUFSIZE)<=0){
            printf("answer_pipe read failed\n");
            printf("%s\n",strerror(errno));
            if(errno == EPIPE) {
                close(answer_pipe[0]);
            }
      }else{
            printf("answer_pipe read successed:%s\n",buffer3);
            sscanf(buffer3, "%s,%d",totsort[++c_use].name,&tmp);
            totsort[c_use].sum_line += tmp;
            totsort[c_use].sum_file += 1;
            totsort[c_use].pid = pid;
       }
```

3.
Core code:

```
sigset( SIGALRM, sUalrm);
   rev = ualarm(inrp, 0);
   if (rev==-1) {
```

```
    perror("ualarm:\n");
    exit(1);
}
```