



गल्स हू कोड एट होम

भूल-भुलैया को डीबग करें
अनप्लग्ड गतिविधि

गतिविधि का संक्षिप्त विवरण

कंप्यूटर विज्ञानी कंप्यूटरों के लिए किसी कार्य विशेष को करने का **कोड**, या निर्देश लिखने का कार्य करते हैं। आपको यह जानकर अचरज हो सकता है कि अधिकांश **प्रोग्रामर**, यानि वे लोग जो कोड लिखते हैं, असल में अपना अधिकांश समय **डीबिंग** में, यानि अपने कोड में समस्याएं ढूँढने और उसे ठीक करने में लगाते हैं! डीबिंग के लिए अटलता और समस्याएं हल करने के नए तरीकों की तलाश करते रहने की इच्छा चाहिए होती है। इस गतिविधि में आपको एक किरदार को एक भूल-भुलैया से होकर ले जाने वाले कोड को डीबग करने का अवसर मिलेगा। आपको अपना खुद का कोड लिखने और अपने किसी मित्र को उसे डीबग करने की चुनौती देने का अवसर भी मिलेगा! इससे पहले कि आप डीबिंग के बारे में और जानना शुरू करें, वुमेन इन टेक स्पॉटलाइट में प्रस्तुत ब्रिटनी वेंगर (Brittany Wenger) पर एक नज़र डालें। 15 वर्ष की आयु में ब्रिटनी ने एक ऐसा टूल बनाया था जो महिलाओं में स्तन कैंसर की पहचान करने में सहायता करता है, ताकि उन्हें शुरुआत में ही उपचार मिल सके।

सामान

- (वैकल्पिक) गेम बोर्ड की छोटी गोटी या वस्तु
- भूल-भुलैया डीबिंग वर्कशीट समाधान

विमेन इन टेक स्पॉटलाइट: ब्रिटनी वेंगर (Brittany Wenger)



जब आपके मन में कोई प्रश्न होता है, तो आप उसका उत्तर किस प्रकार खोजते हैं? यदि आपके पहले स्रोत के पास उत्तर न हो तो आप क्या करते हैं? ब्रिटनी वेंगर (Brittany Wenger) ने स्तन कैंसर से संबंधित महत्वपूर्ण प्रश्नों के उत्तरों की खोज के लिए कोडिंग का सहारा लिया।

जब ब्रिटनी 15 वर्ष की थीं, उनकी कज़िन में स्तन कैंसर होने का पता चला था। अमेरिका में हर 8 में से लगभग 1 महिला अपने जीवनकाल में स्तन कैंसर से प्रभावित होती है। स्तन कैंसर का पता शुरू में ही चल जाने, जब वह एक स्थान में सीमित होता है, से महिला के जीवित बचने की संभावना 100% होती है। ब्रिटनी ने तय किया कि वे कदम उठाएंगी और उन्होंने रोगियों से डेटा एकत्र करके, स्तन का कोई पिंड असाध्य कैंसर का रूप ले चुका है या नहीं (वह मैलिंगनेंट है)

या बेनाइन) यह अनुमान लगाने वाला एक अल्गोरिथ्म (कलन विधि) तैयार किया। उनका अल्गोरिथ्म [क्लाउड4कैंसर \(Cloud4Cancer\)](#) स्तन पिंडों का सही वर्गीकरण करने के मामले में 99.11% की शुद्धता तक पहुंच गया था।

ब्रिटनी वेंगर (Brittany Wenger) का [वीडियो](#) देखकर जानें कि किस प्रकार जिंदगियों को प्रभावित करने वाले बड़े प्रश्नों के उत्तर खोजने के लिए कोड का उपयोग किया जा सकता है।

सोचें

कंप्यूटर विज्ञानी होने का अर्थ केवल कोडिंग में प्रवीण होना नहीं है, यह उससे भी कहीं बढ़कर है। कुछ समय सोचें कि कैसे ब्रिटनी (Brittany) और उनका कार्य उन शक्तियों से संबंधित है जिन्हें बढ़ाने पर कंप्यूटर विज्ञानी फ़ोकस करते हैं, वे हैं: साहस, फिर उठ खड़े होने की क्षमता, रचनाशीलता, और उद्देश्य।



साहस

ब्रिटनी (Brittany) तत्काल उपचार के लिए स्तन कैंसर की शुरुआत में ही पहचान करने के लिए दुनिया भर की महिलाओं की सहायता करने के उद्देश्य से टेक्नॉलॉजी का उपयोग कर सकी।

सोचें, कब आप किसी प्रोजेक्ट या गतिविधि में उम्मीद से भी आगे निकल गए थे। किस बात ने आपको इतनी कड़ी मेहनत के लिए प्रेरित किया?

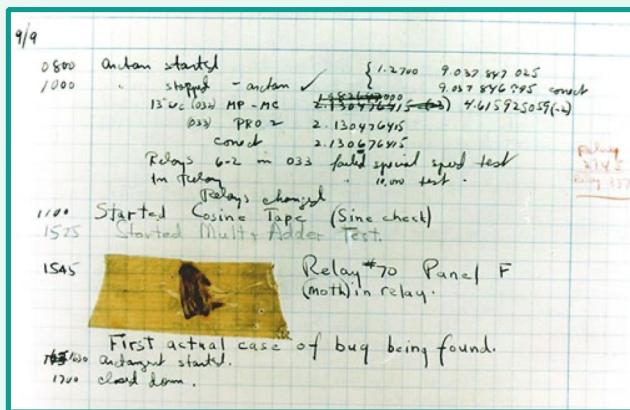
अपने उत्तर किसी परिजन या मित्र से साझा करें। दूसरों को चर्चा में शामिल होने के लिए ब्रिटनी (Brittany) के बारे में और पढ़ने को प्रेरित करें!

चरण 1: डीबगिंग को समझना (2 मिनट)

तब के बारे में सोचें जब आपने किसी समस्या को हल करने की कोशिश की थी, यह स्कूल का कोई कठिन असाइनमेंट हो सकती है, या किसी खोई हुई वस्तु का पता लगाने की कोशिश हो सकती है। क्या आप उसे तुरंत हल कर सके थे? कई बार ऐसा होता है कि जब हम हमारे सामने मौजूद किसी समस्या को हल करने की कोशिश करते हैं तो सफलता से पहले हमें कुछ बार असफलता का मुँह देखना पड़ता है। प्रोग्रामर अपना अधिकांश समय अपने कोड में समस्याएं ढूँढ़ने और उन्हें ठीक करने में लगाते हैं।

किसी कंप्यूटर प्रोग्राम या हार्डवेयर में मौजूद एरर (त्रुटि) को बग (दोष) कहते हैं। कंप्यूटर हार्डवेयर या सॉफ्टवेयर में एरर या बग की पहचान करने और उन्हें हटाने की प्रक्रिया को डीबगिंग (दोषनिवारण) कहते हैं। ये शब्द (बग और डीबगिंग) ग्रेस हॉपर (Grace Hopper) की देन हैं, जो कंप्यूटर विज्ञान के आरंभिक पथ-प्रदर्शकों में से एक थीं। शुरुआती कंप्यूटरों में से एक पर कार्य करते समय, ग्रेस हॉपर (Grace Hopper) की टीम को कंप्यूटर के अंदर एक पतंगा, यानि एक असली भौतिक बग (कीड़ा) मिला जिसके कारण एरर आ रही थी! ग्रेस (Grace) ने अपने जरनल में उस पतंगे को टेप से चिपका दिया और उसे आज इतिहास का पहला कंप्यूटर बग का रिकॉर्ड माना जाता है। यह रिकॉर्ड वॉशिंगटन डी.सी. (Washington D.C.) में स्मिथसोनियन म्यूज़ियम ऑफ अमेरिकन हिस्ट्री (Smithsonian Museum of American History) में प्रदर्शन के लिए रखा हुआ है।

इस गतिविधि में आपको एक किरदार को एक भूल-भुलैया में आरंभ बिंदु से अंत बिंदु तक ले जाने, और इस दौरान उससे सितारे उठवाने वाले कोड को डीबग करने का अवसर मिलेगा। आपको अपना खुद का कोड लिखने और अपने किसी मित्र को उसे डीबग करने की चुनौती देने का अवसर भी मिलेगा!



पहला कंप्यूटर बग का रिकॉर्ड

चरण 2: गेम के नियम (2 मिनट)

गेम का लक्ष्य यह है कि एक किरदार को आरंभ बिंदु से आरंभ करवाते हुए, सारे सितारे उठवाने हैं, और अंत बिंदु पर पहुंचाना है, जिसके लिए हमें निर्देशों में मौजूद सभी बग (यानि समस्याएं) ढूँढ़कर उन्हें ठीक करना है।

चरण 2: गेम के नियम (जारी)

नियम

- आपके किरदार को आरंभ बिंदु से आरंभ करना होगा 
- आपके किरदार को अंत बिंदु पर अंत करना होगा । आपके किरदार के अंत बिंदु तक पहुंचते ही गेम खत्म हो जाएगा और उसके बाद **किसी भी** निर्देश का पालन नहीं होगा।
- आपको अंत बिंदु पर पहुंचने से **★ पहले** सितारे उठाने होंगे।
- किरदार को कोई सितारा उठाने से पहले उसी के स्थान पर होना चाहिए।
- आप निर्देशों को केवल **संपादित** कर सकते या **हटा** सकते हैं। किसी चरण के संपादन का अर्थ है निर्देश में मौजूद किसी भी वैल्यू को हटाकर कोई अन्य वैल्यू रखना या **स्पेलिंग** एरर ठीक करना।
- केवल** इन निर्देशों की अनुमति है। स्पेलिंग और कैपिटल (बिग) और स्मॉल अक्षरों का उपयोग ठीक-ठीक वैसा होना चाहिए जैसा नीचे लिखा है।
 - Move Left (यानि बायें बढ़ें)
 - Move Right (यानि दायें बढ़ें)
 - Move Up (यानि ऊपर बढ़ें)
 - Move Down (यानि नीचे बढ़ें)
 - Repeat [1 to 5] times: (यानि [1 से 5] बार दोहराएँ:)
 - Stop Repeat (यानि दोहराना रोकें)
 - Pick Up Star (यानि सितारा उठाएं)
- आप अपने खुद के निर्देश **नहीं** जोड़ सकते हैं

चरण 3: मान्य निर्देशों के लिए जांचें (2 मिनट)

किसी नई भाषा (जैसे स्पेनिश, फ्रेंच, जर्मन आदि) को सीखने की ही तरह कोडिंग के भी अपने नियम होते हैं। इन विकल्पों में से, उन कोड की पंक्तियों पर गोला बनाएं जो मान्य हैं। याद रखें कि सभी मान्य निर्देश ऊपर वाली सूची में हैं (नियम #6)। याद रखें कि सही स्पेलिंग और कैपिटल (बिग) व स्मॉल अक्षरों का सही उपयोग आवश्यक है!

- | | | | |
|---------------------|-----------------|----------------------|---------------------|
| A. यानि दायें बढ़ें | B. आगे बढ़ें | C. स्टार्ज चुनें | D. 3 बार दोहराएँ: |
| E. 10 बार दोहराएँ: | F. बाईं ओर जाएँ | G. ऊपर चलें | H. दोहराना बंद करें |
| I. सितारा चुनें | J. सितारा हटाएँ | K. लक्ष्य की ओर चलें | L. नीचे जाएँ |

दस्तावेज़ के अंत में दिए गए समाधान पर्चे से अपने उत्तर जांचना न भूलें।

चरण 4: डीबगिंग शुरू करने का समय! (5 मिनट)

नीचे आपको गेम बोर्ड दिखेगा जिसमें आरंभ ध्वज (A, 3) पर है, सितारा (B, 2) पर है, और समाप्ति रेखा (A, 1) पर है। जब हम कोड, यानि निर्देशों को चलाते हैं, तो हमारी यह अपेक्षा होगी कि किरदार आरंभ ध्वज से चलना शुरू करे, सितारा उठाए, और फिर (A, 1) पर स्थित समापन स्थान पर अंत करे। पर हम पाते हैं कि किरदार की वास्तविक क्रियाओं ने सितारा नहीं उठाना और (C, 1) पर अंत करना शामिल है।

	A	B	C
1			
2			
3			

अपेक्षित क्रियाएं	वास्तविक क्रियाएं
<ul style="list-style-type: none"> सितारा उठाएं समापन स्थान (A, 1) पर अंत करना 	<ul style="list-style-type: none"> सितारा नहीं उठाया स्थान (C, 1) पर अंत किया

पहले बताए गए नियमों का उपयोग करते हुए, गलत कोड को ठीक (यानि डीबग) करने में सहायता करें और 'डीबग किया हुआ कोड/निर्देश' शीर्षक के नीचे सही कोड लिखें।

सुझाव: किसी गेम बोर्ड की छोटी सी गोटी या घर की किसी वस्तु का उपयोग किरदार के रूप में करें। जैसे-जैसे आप कोड की हर लाइन एकज़ीक्यूट (निष्पादित) करते जाएं, उसे भूल-भुलैया में निर्देशानुसार बढ़ाते जाएं, इससे आपको किरदार के स्थान का हिसाब रखने में सहायता मिलेगी।

गलत कोड/निर्देश	डीबग किया हुआ कोड/निर्देश
1. ऊपर जाएँ	
2. दाएँ चलें	
3. स्टार्ज चुनें	
4. ऊपर चलें	
5. दाएँ चलें	

संकेत: यहां 2 बग हैं

आप दस्तावेज़ के अंत में दिए गए समाधान पर्चे से अपने उत्तर जांच सकते हैं।

चरण 5: अगली चुनौती से निपटें (5-10 मिनट)

ओह नहीं! एक और बग! नीचे दिए गए निर्देशों को डीबग करने में सहायता करें। नियम जांचना न भूलें।

	A	B	C
1			
2			
3			

अपेक्षित क्रियाएं	वास्तविक क्रियाएं
<ul style="list-style-type: none"> सितारा उठाएं समापन स्थान (B, 2) पर अंत करना 	<ul style="list-style-type: none"> सितारा नहीं उठाया समापन स्थान (B, 2) पर अंत किया

गलत कोड/निर्देश	डीबग किया हुआ कोड/निर्देश
1. ऊपर जाएँ	
2. दाएँ चलें	
3. ऊपर जाएँ	
4. दाएँ चलें	
5. स्टार्ज चुनें	
6. नीचे जाएँ	
7. बाएँ चलें	

संकेत: यहां 2 बग हैं

बोनस: निर्देश, सितारा उठाने और लक्ष्य तक जाने का एक समाधान प्रस्तुत करते हैं, पर ऐसा करने के एक से अधिक मार्ग मौजूद हैं। क्या आप ऐसा एक और कोड या निर्देश समूह लिख सकते हैं जो समान परिणाम देते हैं?

आप दस्तावेज़ के अंत में दिए गए समाधान पर्चे से अपने उत्तर जांच सकते हैं।

चरण 6: लूप (दोहराव) की डीबगिंग करना (5-10 मिनट)

फिर से एक और बग! ज़रा रुको, यह मैप तो पिछली चुनौती वाले मैप जैसा जिखता है। तो सही कोड में उसी मार्ग का पालन होना चाहिए जिसका पालन चुनौती #2 वाले कोड ने किया था, पर इसमें *repeat* कमांड (दोहराव आदेश) का उपयोग होता है। कंप्यूटर विज्ञान में हम इन्हें लूप (Loop) कहते हैं। लूप का उपयोग किसी निर्देश समूह को कई बार दोहराने के लिए किया जाता है। कंप्यूटर को पता हो कि कितनी कोड की पंक्तियाँ दोहरानी हैं यह सुनिश्चित करने के लिए आमतौर पर हम उन निर्देशों को इन्डेन्ट कर देते हैं (यानि थोड़ा दायर्याँ और खिसका देते हैं) जिन्हें हम दुहरवाना चाहते हैं।

	A	B	C
1			
2			
3			

अपेक्षित क्रियाएं	वास्तविक क्रियाएं
<ul style="list-style-type: none"> सितारा उठाएं समापन स्थान (B, 2) पर अंत करना 	<ul style="list-style-type: none"> सितारा नहीं उठाया समापन स्थान (B, 2) पर अंत किया

गलत कोड/निर्देश	डीबग किया हुआ कोड/निर्देश
1. 1 बार दोहराएँ:	
2. ऊपर जाएँ	
3. दोहराना बंद करें	
4. 3 बार दोहराएँ:	
5. दाएँ चलें	
6. दोहराना बंद करें	
7. सितारा चुनें	
8. बाएँ चलें	
9. बाएँ चलें	

संकेत: यहां 3 बग हैं

आप दस्तावेज़ के अंत में दिए गए समाधान पर्चे से अपने उत्तर जांच सकते हैं।

चरण 7: अपना खुद का बनाएं! (10-15 मिनट)

इस बार आपकी बारी है! अपना खुद का ऐसा कोड लिखकर अपने मित्र को चुनौती दें जिसमें अधिकतम 4 बग हों। अपने मित्र के लिए अपेक्षित क्रियाएं बनाम वास्तविक क्रियाएं तालिका भरना न भूलें। इस शीट का उपयोग अपनी उत्तर कुंजी के रूप में करें, और जब आप अपनी चुनौती से संतुष्ट हों, तो अगली शीट में ग़लत कोड, भूल-भुलैया, और क्रिया तालिका कॉपी कर लें और अपने मित्र को चुनौती दें।

सहायक सुझावः पहले सही निर्देश लिखें और फिर किन्हीं 4 या कम लाइनों में बदलाव करके अपने कोड में बग डाल दें।

	A	B	C
1			
2			
3			

अपेक्षित क्रियाएं	वास्तविक क्रियाएं
<ul style="list-style-type: none"> सितारा उठाएं समापन स्थान (,) पर अंत करना 	<ul style="list-style-type: none"> सितारा _____। समापन स्थान (,) पर अंत किया

ग़लत कोड/निर्देश	डीबग किया हुआ कोड/निर्देश
2.	
3.	
5.	
6.	

संकेत: यहां _____ बग हैं

रचनाकार:

	A	B	C
1			
2			
3			

चुनौतीकार:

अपेक्षित क्रियाएं	वास्तविक क्रियाएं
<ul style="list-style-type: none"> सितारा उठाएं समापन स्थान (,) पर अंत करना 	<ul style="list-style-type: none"> सितारा _____। समापन स्थान (,) पर अंत किया

ग़लत कोड/निर्देश	डीबग किया हुआ कोड/निर्देश
2.	
3.	
5.	
6.	

संकेत: यहां बग हैं

चरण 8: अपनी रचना साझा करें! (5 मिनट)

1. अपनी भूल-भुलैया की तस्वीर खींचें या उसे स्कैन करें और अपने मित्र को आपके निर्देश डीबग करने की चुनौती दें!
2. अपनी भूल-भुलैया सोशल मीडिया पर साझा करना न भूलें। @girlswhocode #codefromhome टैग जोड़ दें, और हो सकता है कि हम हमारे अकाउंट पर आपकी रचना दिखाएं!

भूल-भुलैया को डीबग करने के समाधान

चरण 3: मान्य निर्देशों के लिए जांचें

A. दाएँ मुँड़े मान्य कमांड नहीं	B. आगे बढ़ें मान्य कमांड नहीं	C. स्टार्ज़ चुनें स्पेलिंग एरर	D. 3 बार दोहराएं:
E. 10 बार दोहराएं: मान्य कमांड नहीं। दोहराव 1-5 बार किया जा सकता है।	F. दायें मुँड़ें मान्य कमांड नहीं	G. ऊपर चलें	H. दोहराना बंद करें
I. सितारें उठाएं स्पेलिंग एरर	J. सितारा हटाएं मान्य कमांड नहीं	K. लक्ष्य की ओर चलें मान्य कमांड नहीं	L. नीचे जाएँ

चरण 4: डीबगिंग समाधान शुरू करने का समय!

	A	B	C
1			
2			
3			

अपेक्षित क्रियाएं	वास्तविक क्रियाएं
<ul style="list-style-type: none"> सितारा उठाएं समापन स्थान (A, 1) पर अंत करना 	<ul style="list-style-type: none"> सितारा नहीं उठाया स्थान (C, 1) पर अंत किया

गलत कोड/निर्देश	डीबग किया हुआ कोड/निर्देश
1. ऊपर जाएँ	1. ऊपर जाएँ
2. दाएँ चलें	2. दाएँ चलें
3. स्टार्ज चुनें	3. सितारा चुनें
4. ऊपर चलें	4. ऊपर चलें
5. दाएँ चलें	5. बाएँ चलें

चरण 5: अगली चुनौती से निपटें के समाधान

	A	B	C
1			
2			
3			

अपेक्षित क्रियाएं	वास्तविक क्रियाएं
<ul style="list-style-type: none"> सितारा उठाएं समापन स्थान (B, 2) पर अंत करना 	<ul style="list-style-type: none"> सितारा नहीं उठाया समापन स्थान (B, 2) पर अंत किया

शालत कोड/निर्देश	डीबग किया हुआ कोड/निर्देश
1. ऊपर जाएँ	1. ऊपर जाएँ
2. दाएँ चलें	2. ऊपर जाएँ
3. ऊपर जाएँ	3. दाएँ चलें
4. दाएँ चलें	4. दाएँ चलें
5. सितारा चुनें	5. सितारा चुनें
6. नीचे जाएँ	6. नीचे जाएँ
7. बाएँ चलें	7. बाएँ चलें

बोनस: निर्देश, सितारा उठाने और लक्ष्य तक जाने का एक समाधान प्रस्तुत करते हैं, पर ऐसा करने के **एक से अधिक मार्ग मौजूद हैं।** क्या आप ऐसा एक और कोड या निर्देश समूह लिख सकते हैं जो समान परिणाम दे?

एक और प्रस्तावित निर्देश समूह इस प्रकार है: Move Right, Move Right, Move Up, Move Up, Pick Up Star, Move Down, और Move Left. अंतिम दो चरणों का क्रम बदल देने पर भी समाधान मान्य रहेगा। चुनौती #4 रिपीट लूप (repeat loop) का उपयोग करने वाले एक अन्य प्रकार के समाधान की भी छानबीन करेगी।

चरण 6: लूप की डीबगिंग करने के समाधान

	A	B	C
1			
2			
3			

अपेक्षित क्रियाएं	वास्तविक क्रियाएं
<ul style="list-style-type: none"> सितारा उठाएं समापन स्थान (B, 2) पर अंत करना 	<ul style="list-style-type: none"> सितारा नहीं उठाया समापन स्थान (B, 2) पर अंत किया

ग़लत कोड/निर्देश	डीबग किया हुआ कोड/निर्देश
1. 1 बार दोहराएँ:	1. 2 बार दोहराएँ:
2. ऊपर जाएँ	2. ऊपर जाएँ
3. दोहराना बंद करें	3. दोहराना बंद करें
4. 3 बार दोहराएँ:	4. 2 बार दोहराएँ:
5. दाएँ चलें	5. दाएँ चलें
6. दोहराना बंद करें	6. दोहराना बंद करें
7. सितारा चुनें	7. सितारा चुनें
8. बाएँ जाएँ	8. बाएँ चलें
9. नीचे जाएँ	9. नीचे जाएँ

ध्यान दें कि यह समाधान वस्तुतः पिछले समाधान से लंबा है, तो हमें लूप का उपयोग क्यों करना चाहिए? लूप तब उपयोगी होते हैं जब ऐसे कमांड या चरण होते हैं जिन्हें हम बहुत अधिक बार (दो से अधिक बार) दुहरवाना चाहते हैं या ऐसे बहुत से चरण होते हैं जिन्हें हम दुहरवाना चाहते हैं। इस उदाहरण विशेष में हमें चरण कम होते नहीं दिखते, पर अन्य उदाहरणों में इनका उपयोग हमारे कोड को और छोटा बनाने तथा हमारे कोड को पढ़ने वाले किसी भी अन्य व्यक्ति के लिए हमारा कोड और पठनीय (पढ़ने व समझने में आसान) बनाने में निश्चय ही सहायता करेगा।